J Intell Inf Syst
DOI 10.1007/s10844-014-0306-7

Extensions and relationships of some existing lower-bound functions for dynamic time warping

Hailin Li · Libin Yang

Received: 20 November 2012 / Revised: 18 December 2013 / Accepted: 9 January 2014

Abstract Dynamic time warping (DTW) is a state-of-the-art time series similarity measure method, which warps time axes to match the same shape between two time series with different lengths. However, its quadratic time and space complexity is an obstacle to its applications in the large time series data mining. To address this problem, some lower-bound functions for DTW, fast methods to approximately measure the distance between time series, are used to prune the dissimilar objects from time series database so as to retain the candidates for further measuring their similarity with DTW. So far, the existing lower-bound functions for DTW have been widely accepted for time series similarity search and indexing. In this paper, we propose the extensions of two existing lower-bound functions and discuss the relationships among them. The extensions are improved with high tightness and without much time cost. At the same time, we theoretically prove that these extensions satisfy lower-bound requirement and are better than their old versions respectively. The experimental results demonstrate that in most cases the quality of the proposed extensions of lower-bound functions for DTW outperforms the original versions except for a slightly higher time cost.

Keywords Dynamic time warping · Lower-bound function · Similarity measure · Time series data mining

1 Introduction

Similarity (distance) measure is very important for time series data mining. Since many algorithms in time series data mining, such as clustering (Liao 2005; Zhang et al. 2011), classification (Jeong et al. 2011), similarity search (Gullo et al. 2009; Wang and Megalooikonomou 2008; Wang et al. 2010), indexing (An et al. 2005; Keogh 2005) and pattern discovery (Fu 2011; Tang and Liao 2008), must compute the similarity between two time series in advance, there has been more and more attention on this

H. Li (✉) · L. Yang
College of Business Administration, Huaqiao University, Quanzhou 362021, China
e-mail: hailin@mail.dlut.edu.cn

study. It is well known that Euclidean distance is a common method used to measure the similarity between two time series. However, it is often the case that two time series have the same overall component shape, but these shapes do not line up in X-axis (Keogh and Pazzani 2001). In addition, Euclidean distance is sensitive to abnormal points. In other words, very similar time series maybe have large distance because of the distance contribution of the abnormal points. Thereby, Euclidean distance is so rigid and sensitive that in most cases it can not measure the similarity objectively.

To address those problems, dynamic time warping (DTW) (Keogh and Pazzani 2001; Ratanamahatana and Keogh 2004) was proposed, which is a time-warping and more suitable distance function. It allows some alignments in time axes to warp measure the similarity, which means that points respectively deriving from the two time series can match with each other at different times. In this way, it is possible that points with the same shapes match with each other. It also measure the similarity between time series with different lengths and is not sensitive to abnormal points. To align the points best, an optimal warping path should be found, which costs quadratic time and space complexity. Moreover, DTW does not satisfy the triangle inequality as needed by most index methods. Thereby, these two disadvantages make its performance poor when it is used to similarity search and indexing in large time series databases.

Besides improving the efficiency of DTW such as the reduced scope (Sakurai et al. 2005; Salvador and Chan 2007) applied to search the best warping path, there are two ways used to overcome the above mentioned difficulties at least. They are representation techniques for reducing dimension (Boucheham 2010; Li et al. 2011; Mu and Yan 2009) and lower-bound functions (Keogh 2005; Kim et al. 2001; Yi et al. 1998). The former can be regarded as a procedure of data preprocessing. It reduces the dimension of time series so that a few extracted features with a lower dimension representing the raw time series are used to speed up the calculation of DTW. The latter is a group of lower-bound functions for DTW, which can fast prune the dissimilar objects in time series database and retain a small set of similar objects (candidates) to be further measured by DTW. In this way, the time cost of similarity search based on DTW method can be cut down.

So far, only a few lower-bound functions for DTW have been proposed. First, the sum of distances between the maximum (minimum) of a time series and points in the other time series that are larger (smaller) than the maximum (minimum) was defined as a lower-bound function hereafter referred to as LB_Yi proposed by Yi et al. (1998) and Yi and Faloutsos (2000). Second, four features extracted from each time series and the maximum distance of the corresponding feature was reported as another lower-bound distance denoted as LB_Kim proposed by Kim et al. (2001). It is often used to attack the problem of subsequence match under DTW (Kim et al. 2002; Park et al. 2001). Third, LB_Keogh function (Keogh 2005) can be regarded as the Euclidean distance between any parts of the candidate matching sequence not falling within an envelope and the nearest (orthogonal) corresponding section of the envelope. It is one of the most important and popular methods used to time series query (Kremer et al. 2011) and applied to many fields (Keogh 2011; Rakthanmanon et al. 2012). Keogh said that LB_Keogh was the best way to index time series, monitor streams and index shapes. Moreover, LB_Keogh is successfully used to resolve the problems including music retrieval, handwriting retrieval, images indexing, web mining, and so on.

In this paper, we propose extensions of LB_Kim and LB_Keogh and discuss the relationships of the existing lower-bound function. There are two motivations on our work. Firstly, the extension of the lower-bound function is proposed to improve the performance of the old version, which means that the novel lower-bound functions for DTW have greater tightness than the original ones. Secondly, we further analyze the relationships among the lower-bound functions such as LB_Yi, LB_Kim, LB_Keogh and our proposed methods (LB_NKim and LB_NKeogh), which makes the readers understand the relationships these functions have. We deem that LB_Yi is a special case of LB_Keogh and the tightness and pruning power of the extension functions are better than the old versions. Moreover, All the conclusions and explanations are proven theoretically and tested experimentally.

The remainder of the paper is organized as follows. In Section 2, we give the background and related work. The framework of the extensions of lower-bound function for DTW is presented in Section 3. Some experiments perform on different time series datasets in Section 4. In the last section we conclude our work and discuss the future work.

2 Background and related work

DTW is a robust technique often used in the field of time series data mining including clustering, classification and pattern recognition. For better understanding the principle of DTW, the related information about DTW is first given in this section. Meanwhile, since the existing lower-bound functions for DTW are very important for similarity search in time series databases, later we will introduce them in detail.

2.1 Dynamic time warping

Suppose there are two time series Q and C, where $Q = \{q_1, q_2, \cdots, q_n\}$ and $C = \{c_1, c_2, \cdots, c_m\}$. If Euclidean distance function like (1) is used to measure their similarity, the length of time series must be equal, that is $n = m$. In Fig. 1a, it is easy to find that points with the same value of time axes have to rigidly match with each other, which disregards of shape-based match and produces a large distance for the two time series with same shapes.

$$Euc(Q, C) = \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \tag{1}$$

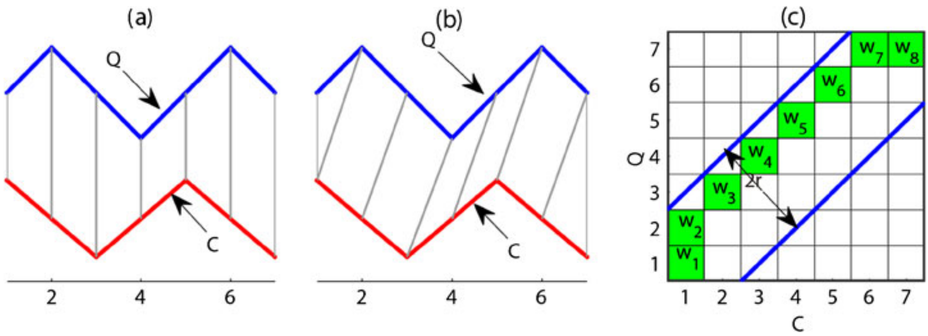

Fig. 1 Similarity between two time series are respectively measured by Euclidean and DTW. **a** shows a rigidly warping path produced by Euclidean. **b** shows that DTW produces a suitable alignment for similarity measure. **c** shows the best warping path in warping matrix and also shows a Sokoe-Chiba band constraint for a warping window

DTW is a warping approach to measure similarity between two time series with equal or unequal lengths. Recently, more attention has been paid to this technique, and its applications have been more widen than before. It adjusts an alignment between two points deriving from different time series and finds an optimal warping path so as to make the two matched points with the same shape as far as possible. As shown in Fig. 1b, DTW can match with the points that have the same shape trends in the two time series, and the distance between the two points matching with each other is a member of the contribution for DTW distance. At the same time, the optimal warping path produces a minimum distance between the two time series. As shown in Fig. 1b and c, each alignment is a member of the optimal warping path.

For two time series Q and C, an matrix D called a warping matrix is constructed in DTW. As shown in Fig. 1c, each cell (i, j) is a member of the warping matrix D and represents an alignment between the points q_i and c_j. Moreover, a sequence of warping matrix elements satisfying the three constraints (such as end-point constraint, monotonicity and continuity) (Keogh and Pazzani 2001) is an whole alignment between Q and C and is called a warping path. Formally, each warping path can be denoted as $W = \{w_1, w_2, \cdots, w_L\}$, where $L \in [max(n, m), n + m - 2]$ and $w_l = (i_l, j_l)$.

Since each element (i, j) is an alignment between q_i and c_j and is associated with a distance between the two points in D, which is often denoted as $d(i, j) = (q_i - c_j)^2$. Thereby, each warping path in warping matrix D can obtain a cumulative distance $E(W)$, that is $E(W) = \sum_{l=1}^{L} d(w_k) = \sum_{l=1}^{L} d(i_l, j_l)$.

DTW must find the optimal warping path which makes its cumulative distance minimal, that is

$$DTW(Q, C) = \underset{\forall W}{min}\{E(W)\} = \underset{\forall W}{min} \sum_{l=1}^{L} d(w_l) \tag{2}$$

To find the best warping path W, dynamic programming approach instead of the brute-force search is used. In detail, a cumulative distance matrix R is constructed and its element $R(i, j)$ represents a cumulative distance between sequences $\{q_1, q_2, \cdots, q_i\}$ and $\{c_1, c_2, \cdots, c_j\}$. $R(i, j)$ satisfies

$$R(i, j) = d(i, j) + \min \begin{cases} R(i, j-1) \\ R(i-1, j-1) \\ R(i-1, j) \end{cases}. \tag{3}$$

In this way, the best warping path W can be obtained, and the minimal distance between Q and C is $R(n, m)$, that is $DTW(Q, C) = R(n, m)$. The elements in warping matrix D contributing to the minimal distance $R(n, m)$ are the members of the best warping path. As shown in Fig. 1c, the green elements (cells) compose the best warping path, and their distance values contribute to the minimal distance $R(n, m)$.

However, DTW must cost a quadratic time complexity to find the best warping path in the warping matrix, which is an undesirable property for time series similarity search. To speed up the calculation of DTW, some constraints such as Sakoe-Chiba Band (Sakoe and Chiba 1978) and the Itakura Parallelogram are applied to reduce the search scope in warping matrix for the best warping path. In other words, these constraints restrict the warping path search in a certain scope of the warping matrix which is called warping window. Moreover, some lower-bound functions for DTW are based on the warping window such as LB_Keogh. Especially, the Sakoe-Chiba Band is often used and defined as $|i_l - j_l| \leq r$ for arbitrary warping path W. As shown in Fig. 1c, the search scope restricted by Sakoe-Chiba Band is on the confines of the two blue lines when $r = 1$.

Generally, the warping window width r can be set according to the percentage λ of the maximal length of the two time series, that is $r = \lceil \lambda * max(n, m) \rceil$, where $\lambda \in [0, 1]$. When $\lambda = 1$, DTW searches the best warping path in the whole warping matrix, which amounts to the traditional DTW. When $\lambda < 1$, DTW searches the best warping path in a local warping matrix. According to the percentage λ, DTW distance can be denoted as

$$DTW_{\lambda}(Q, C) = \min_{\forall W \subseteq D_{\lambda}} \sum_{l=1}^{L} d(w_l), \tag{4}$$

where D_{λ} is the warping window.

In Fig. 2, according to different percentage λ, the warping paths produced by $DTW_{\lambda}(Q, C)$ and the corresponding warping distances are different. Moreover, it shows

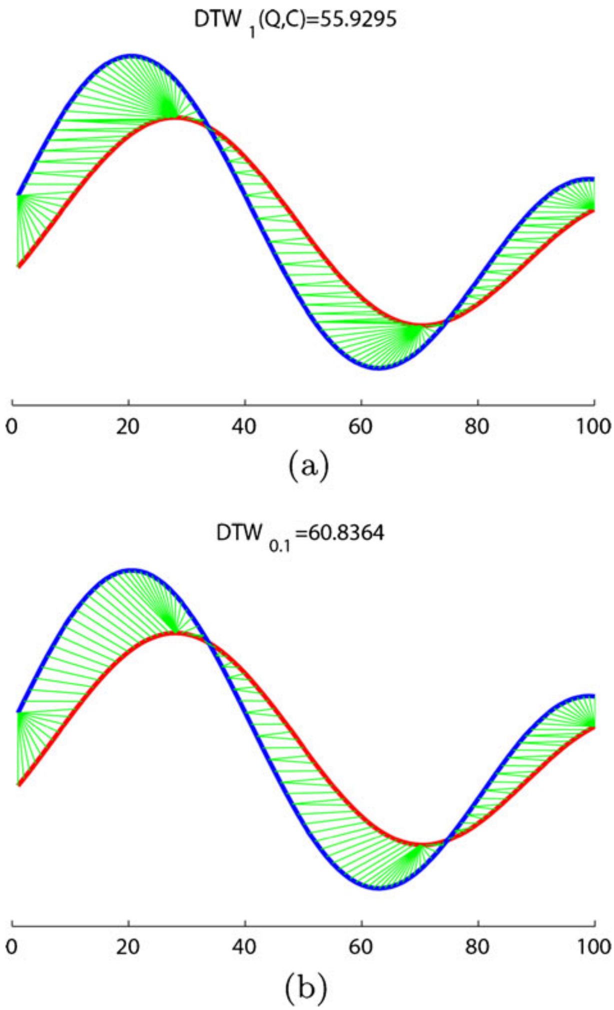

Fig. 2 The alignment between two time series in terms of the warping window width r. **a** shows the alignment when $\lambda = 1$. **b** shows the alignment when $\lambda = 0.1$

that $DTW_{0.1} > DTW_1$, which confirms that DTW can search the optimal warping path in the whole warping matrix and retrieve a minimal distance. It also means that the distance produced by any warping path in the local warping matrix will be equal or greater than that produced by the warping path in the whole warping matrix.

2.2 Existing lower-bound functions for DTW

2.2.1 Tightness of lower-bound function

To remove the disadvantages of DTW, the lower-bound functions for DTW are used. The distance measured by these functions are smaller than that by DTW. Suppose a lower-bound function for DTW is denoted as $LB(Q, C)$, then there is $LB(Q, C) \leq DTW(Q, C)$. If DTW finds the best warping path in a local warping matrix which we often call warping window, there is $LB_\lambda(Q, C) \leq DTW_\lambda(Q, C)$. However, the tightness of a lower-bound function can be measured by (5), where $T \in [0, 1]$ and with larger values generally being better.

$$T = \frac{LB_\lambda(Q, C)}{DTW_\lambda(Q, C)} \tag{5}$$

Note that we sometimes write LB_λ as LB in this paper, e.g., $LB\_Keogh = LB\_Keogh_\lambda$. LB_λ function must be computed more quickly than DTW. If we use DTW to search the most similar object in time series database $S = \{S_1, S_2, \cdots, S_{|S|}\}$ for a query time series C, we need to first test whether $LB_\lambda(S_i, C) \geq \varepsilon$. If it is true, then we deem the current object S_i is dissimilar to C and need not further compute their similarity by DTW because of $DTW_\lambda(S_i, C)$ being also equal to or greater than ε, that is $DTW_\lambda(S_i, C) \geq LB_\lambda(S_i, C) \geq \varepsilon$. In this case, we prune those dissimilar objects. Otherwise, we should retain it as a candidate for further computing the similarity by DTW. So far, there have been three classic lower-bound functions including LB_Yi, LB_Kim and LB_Keogh.

2.2.2 LB_Yi

LB_Yi (Yi et al. 1998) is the sum of every distance between the maximum $max(C)$ (minimum $min(C)$) of one time series C and the point q_i of the other time series Q being larger (smaller) than the maximum (minimum). It can be defined as

$$LB\_Yi(Q, C) = \sum_{q_i > max(C)} d(q_i, max(C)) + \sum_{q_i < min(C)} d(q_i, min(C)). \tag{6}$$

As shown in Fig. 3, the sum of the lengths of the green lines contributes to the overall lower-bound function LB_Yi. In this case, $LB\_Yi(Q, C) = 40.7660$, which is also smaller than $DTW_1(Q, C)$ and means that LB_Yi satisfies lower-bound requirement.

2.2.3 LB_Kim

LB_Kim (Kim et al. 2001) is used to measure two 4-tuple feature vectors $F^q = \left[f_1^q, f_2^q, f_3^q, f_4^q\right]$ and $F^c = \left[f_1^c, f_2^c, f_3^c, f_4^c\right]$ which are derived from different time series Q and C. The 4-tuple feature vector of time series Q is composed of the first point f_1^q, the last point f_2^q, the minimum point f_3^q and the maximum point f_4^q. It means that $f_1^q = q_1$, $f_2^q = q_{|Q|}$, $f_3^q = min(Q)$ and $f_4^q = max(Q)$. The maximum distance of the correspond-

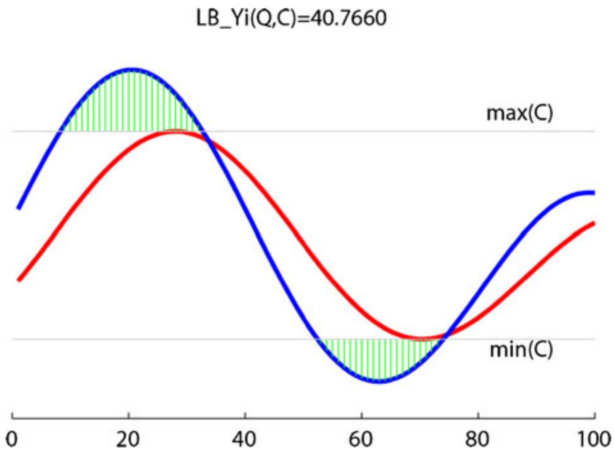

Fig. 3 A visual intuition of the lower bounding function LB_Yi

ing features is reported as the LB_Kim distance between time series Q and C. It can be defined as

$$LB\_Kim(Q, C) = \max_{i=1,2,3,4} d\left(f_i^q, f_i^c\right). \tag{7}$$

As shown in Fig. 4, LB_Kim extracts the four features and retrieves the maximum distance of the corresponding features. In other word, the maximum of the Y-axis difference between the matching points is the distance measured by LB_Kim. Finally, $LB\_Kim(Q, C) = 3.0156$, which is smaller than $DTW_1(Q, C)$ as shown in Fig. 2a. It is also lower-bound for DTW.

2.2.4 LB_Keogh

In contrast to LB_Yi and LB_Kim, the performance of LB_Keogh (Keogh 2005) depends on the warping window, which is based on a local area of the warping matrix *D*. Generally, Sakoe-Chiba Band constraint shown in Fig. 1c is applied to restrict the searching scope of

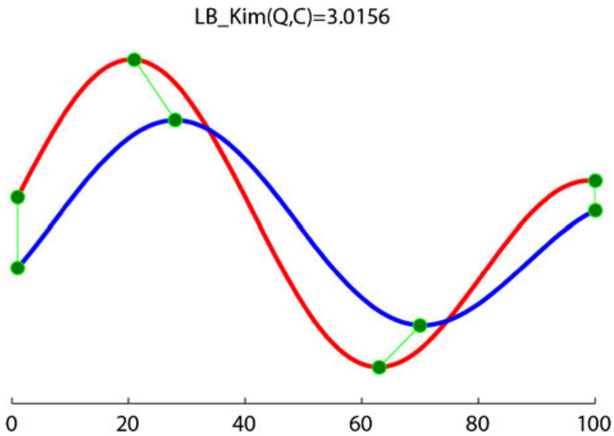

Fig. 4 A visual intuition of the lower bounding function LB_Kim

the warping path. At the same time, the percentage λ about the warping width is often set to be 0.1 (Keogh 2005).

LB_Keogh can be regarded as Euclidean distance between the any part of the candidate matching time series Q not falling within a envelope and the nearest corresponding section of the envelope, where the envelope is composed of two new sequences U and L. That is

$$LB\_Keogh_{\lambda}(Q, C) = \sum_{i=1}^{n} \begin{cases} (q_i - u_i)^2, & if q_i > u_i \\ (q_i - l_i)^2, & if q_i < l_i \\ 0 & otherwise \end{cases}, \quad (8)$$

where $u_i = max(C(i-r : i+r))$ and $l_i = min(C(i-r : i+r))$, r is the warping window width.

As shown in Fig. 5, LB_Keogh_{λ} distance is composed of the green lines that is the distances between the points of time series Q not falling with the envelope (U and L) and the nearest points of the envelope. When $\lambda = 0.1$, $LB\_Keogh_{\lambda}(Q, C) = 49.2788 < DTW_{\lambda}(Q, C) = 60.8364$.

3 Extensions of some existing lower-bound functions

In this section, we analyze some existing lower-bound functions including LB_Yi, LB_Kim and LB_Keogh. The extensions of LB_Kim and LB_Keogh are proposed and the relationships between them are discussed.

3.1 Extension of LB_Kim

LB_Kim uses 4-tuple feature vector to represent the raw time series and retrieves the maximum of the distances between the corresponding features which derive from different time series. However, there are two drawbacks in LB_Kim at least. One is that the maximum regarded as the distance for LB_Kim is often much too small, which makes its tightness small too. The reason is that only one element in the warping matrix D is returned. The other is that LB_Kim is independent of the warping window, which means that it works in the whole warping matrix but not in the local warping matrix (warping window). The former makes LB_Kim be not an excellent lower-bound function because of the smaller tightness.

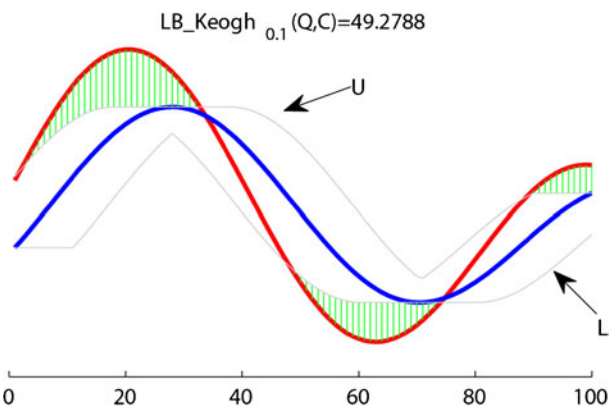

Fig. 5 A visual intuition of the lower bounding function LB_Keogh

As mentioned in the previous section, LB_Kim returns the smallest lower-bound distance value. The latter makes it ineffective and limits the use when a local warping window is applied to the field of similarity search.

To remove the hindrances, we propose a new extension of LB_Kim, called LB_NKim for short. LB_NKim is defined as

$$LB\_NKim_{\lambda}(Q, C) = max\left\{d\left(f_1^q, f_1^c\right), d\left(f_2^q, f_2^c\right), d[Q, max(C)], d[Q, min(C)]\right\}, \tag{9}$$

where

$$\begin{cases} d[Q, max(C)] = \sum\limits_{\substack{q_i > max(C) \\ |i - j_{max}| < r}} d(q_i, max(C)) \\ d[Q, min(C)] = \sum\limits_{\substack{q_i < min(C) \\ |i - j_{min}| < r}} d(q_i, min(C)) \end{cases}, \tag{10}$$

and $j_{max} = \arg\limits_{j} max(c_j)$ and $j_{min} = \arg\limits_{j} min(c_j)$.

$d[Q, max(C)]$ represents the sum of the distances between the maximum $max(C)$ of time series C and the points of the other time series that are greater than the maximum $max(C)$ in the warping window. As shown in Fig. 6a, the green lines are the contribution of LB_NKim when $d[Q, max(C)]$ is the maximum of the four distance features. Similarly, $d[Q, min(C)]$ represents the sum of the distances between the minimum $min(C)$ and the points of the other time series that are less than $min(C)$ in the warping window.

It is easy to prove that:

If $|i - j_{max}| < r$ and $q_i > max(C)$, then there is $d(q_i, max(C)) \leq d(q_i, :)$. However, there being one element $w_{k'}$ of the best warping path $W = \{w_1, w_2, \cdots, w_K\}$ in the warping window at least is an element (i, j') of each row in the warping window because of the three constraints of DTW, that is $w_{k'} \equiv (i, j')$. So $d(q_i, max(C)) = d(i, j_{max}) \leq d(i, j) = d(w_{k'})$. For different i, we have

$$\sum_{\substack{q_i > max(C) \\ |i - j_{max}| < r}} d(q_i, max(C)) \leq \sum_{k' \doteq j'} d\left(w'_k\right) \leq DTW_{\lambda}(Q, C),$$

which means that $d[Q, max(C)] \leq DTW_{\lambda}(Q, C)$. For $d[Q, min(C)]$, we can derive the similar conclusion.

In LB_NKim, the last two elements of the 4-tuple feature vector for time series Q are respectively extended to be a sequence of points greater than the maximum $max(C)$ and to be a sequence of points smaller than the minimum $min(C)$. In this way, LB_NKim not only increases the number of alignments and also enlarges the lower-bound distance value. As shown in Fig. 6b, when $\lambda = 1$, LB_NKim and LB_Kim search the contributed elements in the whole warping matrix. It is obvious that $LB\_NKim_1(Q, C)$ is greater than $LB\_Kim(Q, C)$. In other words, LB_Kim contains only one element of the whole warping matrix. The element is the one of the four feature counterparts $\{(1, 1), (n, m), (i_{max}, j_{max}), (i_{min}, j_{min})\}$. However, LB_NKim possibly contains more than one element of the whole warping matrix, as shown in Fig. 6b, the green lines denote more than one element of the whole warping matrix existed in LB_NKim.

Comparing LB_NKim in (7) with LB_Kim in (9), it is obviously that $LB\_Kim \leq LB\_NKim_1$. The main reason is that the distances of the last two corresponding features in 4-tuple feature vector in LB_Kim are equal to or less than that in LB_NKim. However, when comparing the distances of the last two group correspond features of LB_NKim in (10) with LB_Yi in (6) under $\lambda = 1$, $d[Q, max(C)]$ and $d[Q, min(C)]$ are respectively

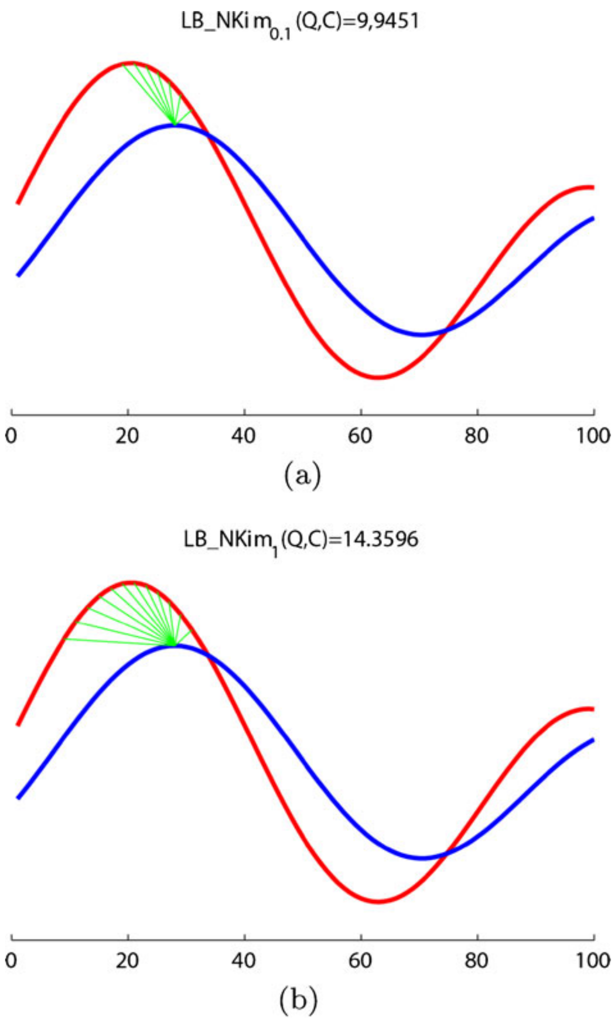

Fig. 6 A visual intuition of the lower bounding function LB_NKim. **a** and **b** respectively show the contributions of LB_NKim distance according to $\lambda = 1$ and $\lambda = 0.1$

equivalent to two terms of LB_Yi, $\sum\limits_{q_i > max(C)} d(q_i, max(C))$ and $\sum\limits_{q_i < min(C)} d(q_i, min(C))$. Thereby, if $d[Q, max(C)]$ or $d[Q, min(C)]$ is the contribution of LB_NKim_1 distance, then $LB\_NKim_1 \leq LB\_Yi$. However, if $\lambda < 1$, there is not a suitable way to compare them because LB_Kim and LB_Yi only depend on the whole warping matrix instead of the local one.

3.2 Extension of LB_Keogh

From Fig. 5, we know that LB_Keogh is a function relying on the upper boundary and the lower boundary that are the maximum sequence U and the minimum sequence L of a time series. They form a bounding envelope that encloses the time series from above and below.

Moreover, the formation of the two sequences U and L depends on the local warping matrix which is often defined by Sakoe-Chiba Band.

To enlarge the tightness of LB_Keogh and make it better for similarity search, we extend (8) to (11). We denote the new extension of LB_Keogh as LB_NKeogh.

$$LB\_NKeogh_{\lambda}(Q, C) = \sum_{i=1}^{n} \begin{cases} (q_i - u_i)^2, & if q_i > u_i \\ (q_i - l_i)^2, & if q_i < l_i \\ \min\limits_{b \le j \le e} (q_i - c_j)^2 & otherwise \end{cases} \tag{11}$$

where $b = max(1, i - r)$ is the beginning column coordinate and $e = min(m, i + r)$ is the ending column coordinate in the warping window.

LB_NKeogh adds a non-zero term to enlarge the tightness of LB_Keogh. It means that if the point q_i falls in the envelope, then we consider the minimum one of the distances between the point q_i and any points of time series C whose column coordinates are within the warping window. As shown in Fig. 7 and compared with Fig. 5, LB_NKeogh adds the warping elements locating in the two time series. So it is easy to conclude that $LB\_Keogh_{\lambda} \le LB\_NKeogh_{\lambda}$.

Now we proof that LB_NKeogh is a lower-bound function for DTW.

Suppose there is an optimal warping path $W = \{w_1, w_2, \cdots, w_K\}$ existed in the warping window according to the percentage λ, where $K \ge n$. From the constraints of DTW we know that the optimal warping path interacts with each row $(i, b : e)$ of the warping window at least at one point w'_i. Moreover, since u_i is the maximum point of the sequence $C(b : e)$, there is $d(q_i, u_i) \le d(q_i, c_j)$ for any $j \in [b, e]$, which means that $d(q_i, u_i) \le d(w_{i'})$ because w'_i is an element of the row $(i, b : e)$. Similarly, for l_i, we can derive a same conclusion, that is $d(q_i, l_i) \le d(w_{i'})$. However, when q_i falls in the range $[l_i, u_i]$, $\min\limits_{b \le j \le e} (q_i - c_j)^2$ is the minimum one of the distances between the point q_i and each point of the sequence $C(b : e)$. It means that $\min\limits_{b \le j \le e} (q_i - c_j)^2 \le d(q_i, j)$ for any $j \in [b, e]$. So we also have $\min\limits_{b \le j \le e} (q_i - c_j)^2 \le d(w_{i'})$ because $w_{i'}$ is also an element of the row $(i, b : e)$ in the warping window. Thereby, for any i, there is one choice from the above three cases, which means

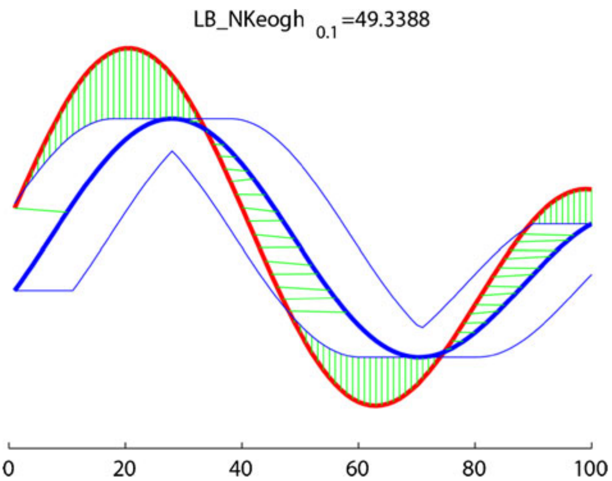

Fig. 7 A visual intuition of the lower bounding function LB_NKeogh

that $LB\_NKeogh_{\lambda}(Q, C) \leq \sum_{i=1}^{n} d(w_{i'}) \leq \sum_{k=1}^{K} d(w_k) = DTW_{\lambda}(Q, C)$. Therefore, LB_NKeogh is a lower-bound function for DTW.

4 Relationships among lower-bound functions

In previous work (Kim et al. 2001; Yi et al. 1998; Keogh 2005), the relationships among different lower-bound functions for DTW have not been discussed. In this section, we analyze their relationships for better applications in the field of time series data mining.

LB_NKim and LB_NKeogh are the extensions of LB_Kim and LB_Keogh respectively. First, in most cases, there is $LB\_Kim \leq LB\_NKim_1$. In other words, if their distances derive from one of the last two terms in the lower-bound functions, in this case there is $LB\_Kim \leq LB\_NKim_1$. Moreover, in most cases their distances do derive from one of the last two terms so that $LB\_Kim \leq LB\_NKim_1$ is true. Second, it is obvious that $LB\_Keogh_{\lambda} \leq LB\_NKeogh_{\lambda}$ is true for any λ because of a non-zero term added in the LB_NKeogh_{λ}. Third, LB_NKim_1 is often smaller than LB_Yi because $d(Q, max(C))$ and $d(Q, min(C))$ are equivalent to the two terms of LB_Yi respectively and LB_NKim_1 only choose the maximal one as the last distance. On the contrary, if $d[Q, max(C)]$ or $d[Q, min(C)]$ is not the contribution of LB_NKim, $LB\_NKim_1 \leq LB\_Yi$ is possible false.

It is interesting that when $\lambda = 1$, LB_Keogh_1 is equivalent to LB_Yi. In LB_Keogh_{λ}, the upper boundary and the lower boundary are respectively the maximum and minimum sequences of time series C in the warping windows. However, when the warping window is extended to the whole warping matrix, namely, $\lambda = 1$, the upper boundary and lower boundary will become the maximum and minimum of the time series. It means that the upper boundary and lower boundary in Fig. 5 are respectively changed into the horizontal lines $max(C)$ and $min(C)$ in Fig. 3. Thereby, LB_Keogh_{λ} will degenerate to LB_Yi.

Comparing LB_NKim_{λ} with LB_Kim, we know that their first and second candidate distances are the same, that are $d\left(f_1^q, f_1^c\right)$ and $d\left(f_2^q, f_2^c\right)$. However, the last two candidate distances are different, and there exists a special relationship. The last candidate distances in LB_Kim are equal to or less than those in LB_NKim, $d(i_{max}, j_{max}) \leq d[Q, max(C)]$ and $d(i_{min}, j_{min}) \leq d[Q, min(C)]$. All those cause that $LB\_NKim_1(Q, C)$ is equal to or greater than $LB\_Kim(Q, C)$ when the last two candidate distances are the contribution of the final distance. At the same time, $d[Q, max(C)]$ and $d[Q, min(C)]$ in LB_NKim_{λ} can be extended to the two terms of LB_Yi, and they are equivalent to each other respectively when $\lambda = 1$ in LB_NKim_{λ}, that is

$$\begin{cases} d[Q, max(C)] = \sum\limits_{q_i > max(C)} d(q_i, max(C)) \\ d[Q, min(C)] = \sum\limits_{q_i < min(C)} d(q_i, min(C)) \end{cases}.$$

Similarly, the two terms of LB_Yi can be deduced to the sum of the two terms of LB_Keogh_{λ}, and the corresponding term of LB_Yi is equivalent to the sum of the corresponding term of LB_Keogh_{λ} when $\lambda = 1$. Finally, LB_Keogh and additional term compose the novel lower-bound function LB_NKeogh_{λ}.

It is obvious that there exist some relationships among the lower-bound functions including LB_Kim, LB_NKim_λ, LB_Yi, LB_Keogh_λ and LB_NKeogh_λ. There are four following relationships among them at least.

(1) LB_Kim is extended to LB_NKim_λ, and in most cases $LB\_Kim(Q, C)$ is smaller than $LB\_NKim_1(Q, C)$ in terms of different contributors of the terms in lower-bound functions.
(2) LB_Yi is a special case of $LB\_Keogh_\lambda(Q, C)$, which means that $LB\_Yi(Q, C)$ is equivalent to $LB\_Keogh_1(Q, C)$.
(3) The last two distance measurements in the vector of $LB\_NKim_1(Q, C)$ are equivalent to the terms of $LB\_Yi(Q, C)$. Since LB_Yi is the sum of two distance measurements but part of LB_NKim is the maximum of the two, $LB\_NKim_1(Q, C)$ is sometimes equal or less than $LB\_Yi(Q, C)$. Moreover, since there is $LB\_Yi = LB\_Keogh_1$, we sometimes have $LB\_NKim_1(Q, C) \leq LB\_Keogh_1$ according to the second relationship.
(4) $LB\_Keogh_\lambda(Q, C)$ and newly increased term compose the extension of lower-bound function $LB\_NKeogh_\lambda(Q, C)$. Meanwhile, $LB\_Keogh_\lambda(Q, C) \leq LB\_NKeogh_\lambda(Q, C)$.

5 Experimental evaluation

Although we have analyzed the extensions of the lower-bound functions and their relationships in theory, some elaborate experiments also should be carried out to compare their performance and further state that the extension is an improved version of the corresponding lower-bound function for DTW. In this section, besides an introduction of time series datasets used in our experiments, comparison experiments among the lower-bound functions for DTW are designed, including tightness, pruning power and time cost.

5.1 Experimental datasets

To objectively show the performance of the lower-bound functions, our experiments perform on a group of different UCR databsets (Keogh et al. 2011), a stock time series (Stock data web page. http://www.cs.ucr.edu/~wli/FilteringData/stock.zip) and random walk data. The UCR datasets (Keogh et al. 2011) are very classic for time series data mining including classification, clustering, pattern discovery, similarity search and so on. In our experiments, we choose 20 time series datasets at will as shown in Table 1. For comparison of the performance of the lower-bound functions in terms of different lengths, the long stock time series of length 2119415 is used to extract some groups of time sequences with different lengths. Figure 8 shows a sequence of length 10000 which is extracted from the stock time series at random. The random walk datasets can be created according to different lengths, which are used to test the efficiency of the methods.

In Table 1, it is easy to know that there are 20 different time series datasets whose sizes, class numbers and lengths are various. Moreover, each dataset is composed of a testing set and a training set. Time series from the two sets are strongly different so as to better investigate the performance of the lower-bound functions.

Table 1 Information of UCR datasets

ID	Name	Class number	Training size	Testing size	Length
1	Adiac	37	390	391	176
2	Beef	5	30	30	470
3	CBF	3	30	900	128
4	Coffee	2	28	28	286
5	ECG200	2	100	100	96
6	FISH	7	175	175	463
7	FaceAll	14	560	1,690	131
8	FaceFour	4	24	88	350
9	Gun_Point	2	50	150	150
10	Lighting2	2	60	61	637
11	Lighting7	7	70	73	319
12	OSULeaf	6	200	242	427
13	OliveOil	4	30	30	570
14	SwedishLeaf	15	500	625	128
15	Trace	4	100	100	275
16	Two_Patterns	4	1,000	4,000	128
17	Synthetic_Control	6	300	300	60
18	Wafer	2	1,000	6,174	152
19	50Words	50	450	455	270
20	Yoga	2	300	3,000	426

5.2 Comparison of tightness

Tightness is an important performance index for a lower-bound function. In this experiment, we let the lower-bound functions participate in the comparison of tightness, including LB_Kim, LB_NKim_λ, LB_Yi, LB_Keogh_λ and LB_NKoegh_λ. At the same time, since LB_Kim and LB_Yi are independent of the warping window width so that we first take two kinds of experiments on tightness according to the different values of the percentage λ. Next, we also investigate the tightness of the lower-bound function in terms of different lengths and different warping window widths.

The first experiment on time series datasets is made to evaluate the tightness of all the lower-bound functions for DTW when they are based on the whole warping matrix. In

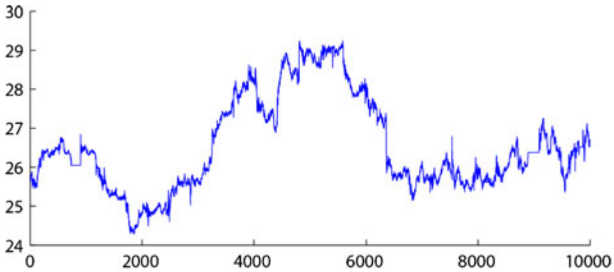

Fig. 8 A sequence of length 10000 extracted from the long stock time series

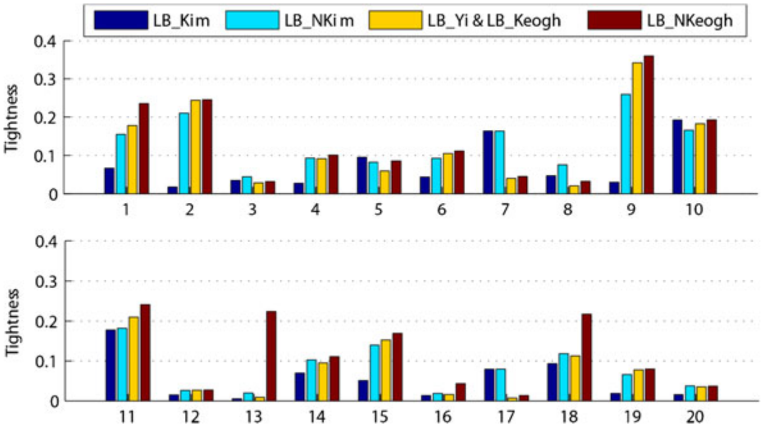

Fig. 9 Tightness of the lower-bound functions when $\lambda = 1$

other words, the experiments are performed on time series datasets for all the lower-bound functions when $\lambda = 1$. We know that LB_NKim_1, LB_Keogh_1 and LB_NKeogh_1 search the best warping path in the whole warping matrix, which is similar to LB_Kim and LB_Yi. In this experiment, we randomly choose a percentage (here is 0.2) of testing time series and training time series as the testing set and the training set respectively. We compare each time series in the testing set with every time series in the training set, and the averaged result is returned when all the time series in testing are finished to compute the lower-bound function. The results of each lower-bound function in different time series sets are shown in Fig. 9.

In Fig. 9, we at least obtain two messages that the tightness of LB_Yi is the same to LB_Koegh_1, and they are sometimes greater than LB_NKim_1 because of the different contributed terms in their distance functions. LB_NKim and LB_NKoegh are the extensions of LB_Kim and LB_Keogh respectively. Their results tell us that the tightness of

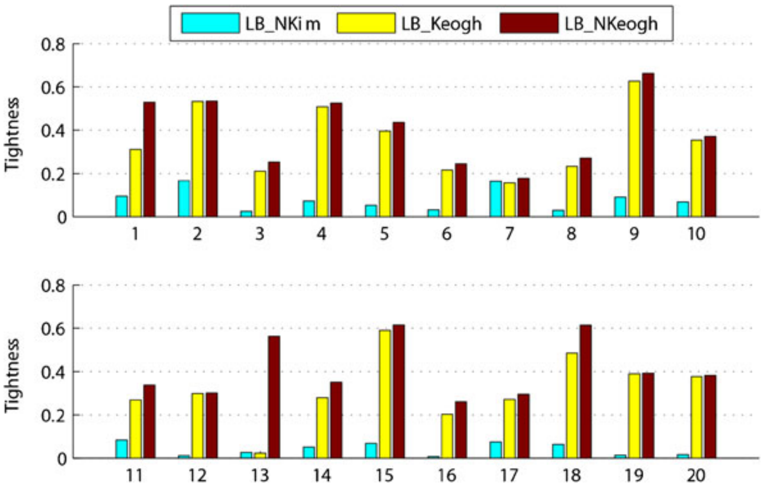

Fig. 10 Tightness of the three lower-bound functions when $\lambda = 0.1$

Table 2 The averaged results of the tightness for different λ are shown

	LB_Kim	LB_NKim	LB_Yi	LB_Keogh	LB_NKeogh
$\lambda = 1$	0.0647	0.1900	0.1109	0.1109	0.1415
$\lambda = 0.1$	$*$	0.0886	$*$	0.3306	0.4000

LB_NKim is better than that of LB_Kim except for the two datasets (No.5 and No.10), and the tightness of LB_NKeogh_1 is always better than that of LB_Keogh_1. Actually, the first three relationships described in previous section are testified in this experiment.

In terms of the factor λ being less than 1, the second experiment is the tightness comparison of the lower-bound functions including LB_NKim, LB_Keogh and LB_NKeogh. The way of this experiment is the same to that of the first experiment in this section. The results of the three lower-bound functions in different time series datasets are shown in Fig. 10, which demonstrate that LB_NKeogh is the best in all.

From the above two experiments, we know that different results are obtained in the different datasets. We averaged the results over all the datasets and got the averaged results as shown in Table 2 that also shows that LB_NKim and LB_NKeogh are respectively tighter than other methods for different λ.

The third experiment is the tightness comparison of the lower-bound functions in terms of the different lengths. We extract six groups of time sequences from the long time series according to the different lengths [16, 32, 64, 128, 256, 512]. To better compare the performance of the extensions and the old versions, , we set $\lambda = 1$ for LB_Kim in the following experiments. The comparison results based on $\lambda = 0.1$ as shown in Fig. 11 state that the tightness of four lower-bound functions including LB_NKim, LB_Keogh and LB_NKeogh will degrade as the length of sequence increases. Moreover, we also can obtain the conclusion that the tightness of LB_NKim and LB_NKeogh is always greater than or equal to their old versions LB_Kim and LB_Koegh respectively.

The last experiment is the tightness comparison of the lower-bound function in terms of the different warping window widths which are described by the percentage λ. We use the group of the sequences of length 1024 as the experimental data and let every two sequences compute the lower-bound functions according to different λ. The results as shown in Fig. 12 tell us that LB_Keogh and LB_NKeogh are decreasing as the warping width increases. However, the extension LB_NKim is increasing. Although LB_Kim is independent of λ, we compare it with LB_NKim. The results tell us that the smaller the λ is, the more similar LB_NKim and LB_Kim will be. The reason is that the smaller the λ is, the more the

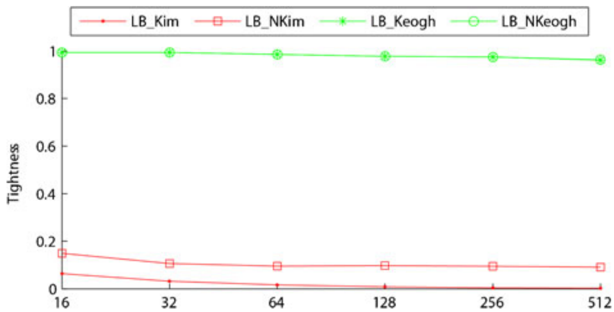

Fig. 11 Tightness comparison of the lower-bound functions ($\lambda = 0.1$) as the length of sequence increases

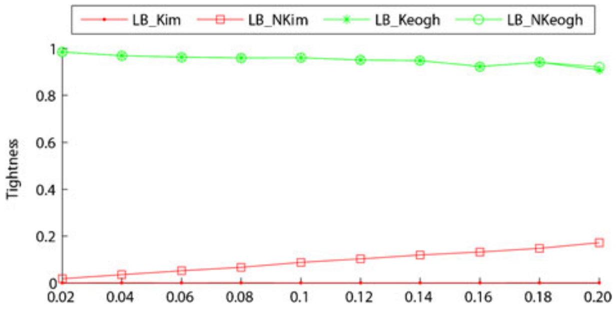

Fig. 12 Tightness comparison of the lower-bound functions as the warping window width increases

possibility of $d[Q, max(C)]$ and $d[Q, min(C)]$ in LB_NKim be equal to $d(i_{max}, j_{max})$ and $d(i_{min}, j_{min})$ in LB_Kim respectively will be. At the same time, the results tell us that the tightness of the extension are greater than or equal to that of the old version as the warping window width increases.

5.3 Comparison of pruning power

Pruning power is another performance index of the lower-bound functions. In time series dataset $S = \{S_1, S_2, \cdots, S_{|S|}\}$, if we need to search the most similar time series for a query time series C by DTW_λ, we can first test whether $LB_\lambda(S_i, C) \geq \varepsilon$. The reason is that the computation of the lower-bound functions is faster than DTW and they roughly prune the dissimilar objects in the dataset S before using DTW to search the most similar time series. If the inequality $LB_\lambda(S_i, C) \geq \varepsilon$, then we regard the retained object S_i as the one dissimilar to C and need not further compute their similarity by DTW. In this case, we prune most of

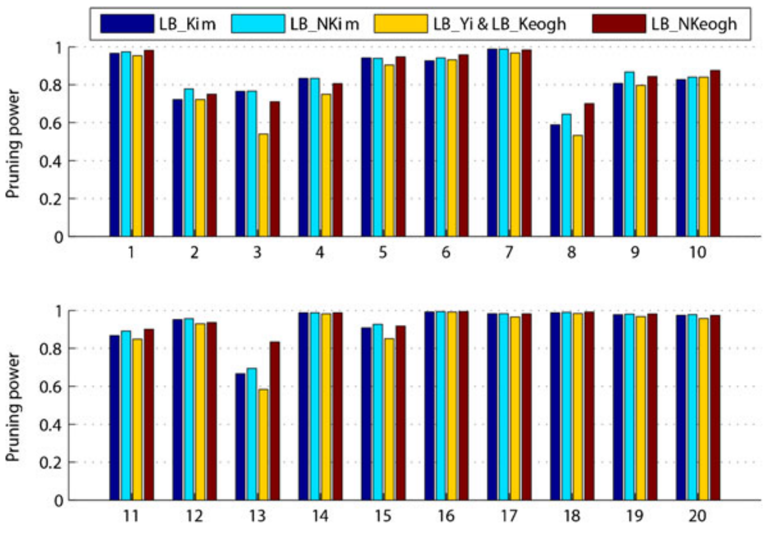

Fig. 13 Pruning power comparison of the lower-bound functions when $\lambda = 1$

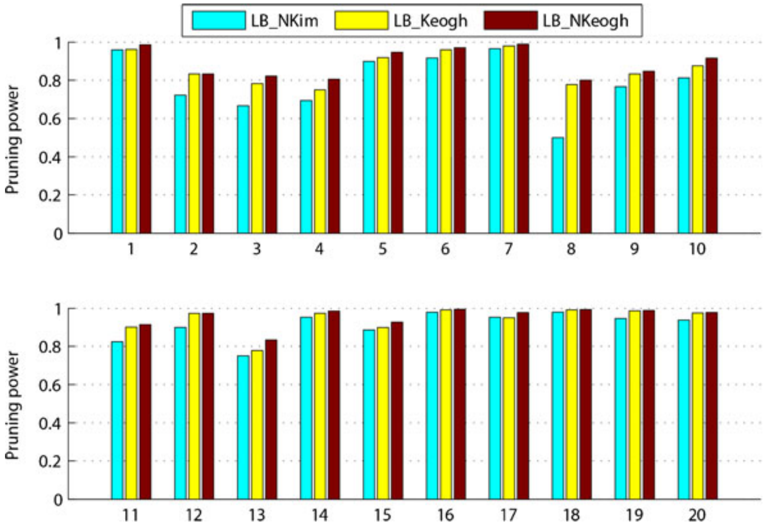

Fig. 14 Pruning power comparison of the lower-bound functions when $\lambda = 0.1$

the dissimilar objects in S. Moreover, we regard the retained object as a candidate for further measuring the similarity by DTW. Then the pruning power can be defined as

$$\texttt{Pruning power} = \frac{\texttt{Number of the dissimilar objects expeled by LB}}{\texttt{Total number of all sequences in the dataset}}. \tag{12}$$

We also use the UCR datasets to investigate the pruning power of the lower-bound functions. The ways to take the experiments on the pruning power are like the first and second experiments on tightness of the lower-bound functions. The two experiments are based on $\lambda = 1$ and $\lambda = 0.1$ respectively. At the same time, we also give the distinct comparison about the pruning power between the original lower-bound function and its extension.

The results about pruning power for each lower-bound function are shown in Fig. 13 when $\lambda = 1$. It is easy to know that the pruning power of the extension is greater than that of the old version when $\lambda = 1$. The same conclusions about pruning power appear at

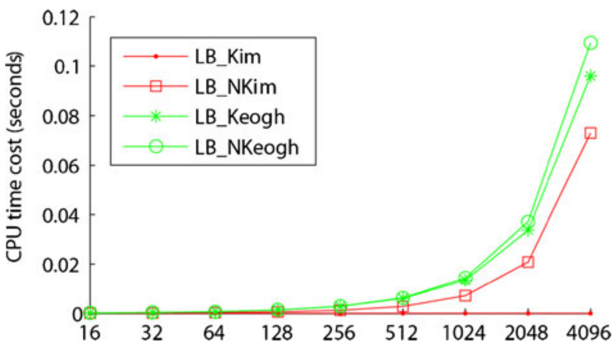

Fig. 15 The CPU time comparisons for the lower-bound functions according to different-length sequences

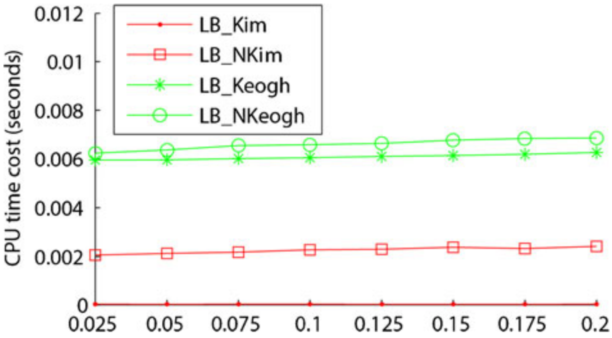

Fig. 16 The CPU time comparisons for the lower-bound functions according to different widths of the warping windows

the experiment under $\lambda = 0.1$ as shown in Fig. 14. The comparison results tell us that in most cases the extensions are better than their original ones at the pruning power. In this experiment, we can state that LB_NKeogh is the best.

5.4 Comparison of calculation time cost

The lower-bound function must be faster than DTW, which had been proven. In this section, we make two experiments to see how much time the lower-bound functions cost. The first one is the comparison of CPU calculation time for different lower-bound functions in terms of different-length sequences. A long random walk sequence of length 10,000,000 are applied to the subsequence matching for the comparison of CPU calculation time. The lengths of query sequences are respectively 16,32,64,128,256,512,1024,2048 and 4096. The lower-bound functions are used to measure the similarity between the subsequence and the query one. In this way, the time cost can be accumulated for the subsequence matching. The final result is an averaged value of the accumulated time cost. The averaged results are shown in Fig. 15 when $\lambda = 0.1$, which indicates that the CPU calculation time increases as the length of sequence increases. The results tell us that the time cost of the extension LB_NKeogh is slightly higher than that of LB_Keogh. Actually, their time complexity is linear to the length (n) of the sequence, that is $O(kn)$. However, the extra time cost is caused by the computation of the non-zero term in (11), which makes the k different for the two methods. Similarly, The time cost of LB_NKim is higher than LB_Kim but lower than the other two methods.

We implement another experiment to compare the CPU calculation time in terms of different widths of the warping window. When λ is set to be 0.1 and the query length is 512, the results of time calculation for the lower-bound functions are shown in Fig. 16. It is easy to find that the CPU time is increasing as the width of the warping window increases. Moreover, the conclusions and reasons about the CPU time are the same as those in the different-lengths experiment.

6 Conclusion

Since dynamic time warping has been a robust method to measure the similarity between the time series with a high time cost, some lower-bound techniques for DTW were proposed to fast prune the dissimilar objects in the databases for a query time series. In this paper, we extend two existing lower-bound functions, LB_NKim and LB_NKeogh. We also discuss the relationships between those lower-bound functions including the two extensions and the existing methods. In theory, we prove that the extensions are really lower-bound for DTW and their tightness are better than the old versions. Moreover, experimental results indicate that in most cases the tightness and pruning power of the proposed extensions are better than the existing lower-bound function except for the extra time cost. It is very important that we discuss the relationships among the existing lower-bound functions, which to the best of our knowledge the previous work had never been revealed. The relationships are concluded as follows:

(1) LB_Kim can be extended to LB_NKim_λ, and LB_Kim is often smaller than LB_NKim_1.
(2) The last two distance measurements in the vector of LB_NKim_1 are equivalent to the terms of LB_Yi, which causes that LB_NKim_1 is sometimes less than LB_Yi.
(3) LB_Yi is a special case of LB_Keogh_1, which means that LB_Yi is equivalent to LB_Keogh_λ when $\lambda = 1$, that is $LB\_Yi = LB\_Keogh_1$.
(4) LB_Keogh_λ and a new non-zero term compose the novel lower-bound function LB_NKeogh_λ. Meanwhile, there is $LB\_Keogh_\lambda \leq LB\_NKeogh_\lambda$.

LB_NKeogh has a little more time cost than LB_Keogh. We must study to reduce the time cost of LB_NKeogh in the future. Meanwhile, We know that LB_Keogh is one of the most popular methods used to indexing and similarity search in the field of time series data mining. It has already been successfully used in many places. LB_NKeogh as an improved version of LB_Keogh can attempt to resolve the problems in the related applications.

Acknowledgments This work has been partly supported by the National Natural Science Foundation of China (61300139) and the Society and Science Planning Projects in Fujian (2013C018). We also would like to acknowledge Prof. Eamonn Keogh for the datasets.

References

An, J., Chen, Y.P.P., Chen, H. (2005). DDR: an index method for large time-series datasets. *Information Systems*, *30*(5), 333–348.

Boucheham, B. (2010). Reduced data similarity-based matching for time series patterns alignment. *Pattern Recognition Letters*, *31*(7), 629–638.

Fu, T.C. (2011). A review on time series data mining. *Engineering Applications of Artificial Intelligence*, *24*(1), 164–181.

Gullo, F., Ponti, G., Tagarelli, A., Greco, S. (2009). A time series representation model for accurate and fast similarity detection. *Pattern Recognition*, *42*(11), 2998–3014.

Jeong, Y.S., Jeong, M.K., Omitaomu, O.A. (2011). Weighted dynamic time warping for time series classification. *Pattern Recognition*, *44*(9), 2231–2240.

Keogh, E. (2005). Exact indexing of dynamic time warping. In *Proceedings of the 28th international conference on very large data bases*, (pp. 358–380).

Keogh, E. (2011). Welcome to the LB_Keogh homepage. http://www.cs.ucr.edu/~eamonn/LB_Keogh.htm.

Keogh, E., & Pazzani, M.J. (2001). Derivative dynamic time warping. In *Proceedings of the first SIAM international conference on data mining*, (pp. 1–11).

Keogh, E., Xi, X., Li, W., et al. (2011). Welcome to the UCR time series classification/clustering page. http://www.cs.ucr.edu/eamonn/time_series_data/.

Kim, S.W., Park, S.H., Chu, W.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).

Kim, S.W., Yoon, J.H., Park, S.H., Kim, T.H. (2002). Shape-based retrieval of similar subsequences in time-series databases. In *Proceedings of the 2002 ACM symposium on applied computing*, (pp. 438–445).

Kremer, H., Günnemann, S., Ivanescu, A., Assent, I., Seidl, T. (2011). Efficient processing of multiple DTW queries in time series databases. In *Proceeding of the 23nd international conference on scientific and statistical database management*, (pp. 150–167).

Li, H., Guo, C., Yang, L. (2011). Similarity measure based on piecewise linear approximation and derivative dynamic time warping for time series mining. *Expert Systems with Applications*, *38*(12), 14732–14743.

Liao, T.W. (2005). Clustering of time series data—a survey. *Pattern Recognition*, *38*(11), 1857–1874.

Mu, B., & Yan, J. (2009). An adaptive filter technique for time series search. In *Proceedings of the 2rd international conference on business intelligence and financial engineering*, (pp. 283–287).

Park, S.H., Kim, S.W., Chu, W.W. (2001). Segment-based approach for subsequence searches in sequence databases. In *Proceedings of the 2001 ACM symposium on applied computing*, (pp. 248–252).

Rakthanmanon, T., Campana, B., Mueen, A., et al. (2012). Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD conference on knowledge discovery and data mining*, (pp. 262–270).

Ratanamahatana, C.A., & Keogh, E. (2004). Everything you know about dynamic time warping is wrong. *SIGKDD workshop on mining temporal and sequential data*.

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

Sakurai, Y., Yoshikawa, M., Faloutsos, C. (2005). FTW: fast similarity search under the time warping distance. In *Proceedings of the 2005 principles of database systems*, (pp. 326–337).

Salvador, S., & Chan, P. (2007). FastDTW: toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, *11*(5), 561–580.

Tang, H., & Liao, S.S. (2008). Discovering original motifs with different lengths from time series. *Knowledge-Based Systems*, *21*(7), 666–671.

Wang, Q., & Megalooikonomou, V. (2008). A dimensionality reduction technique for efficient time series similarity analysis. *Information Systems*, *33*(1), 115–132.

Wang, Q., Megalooikonomou, V., Faloutsos, C. (2010). Time series analysis with multiple resolutions. *Information Systems*, *35*(1), 56–74.

Yi, B.K., & Faloutsos, C. (2000). Fast time sequences indexing for arbitray L_p norms. In *Proceedings of the 26th international conference on very large data bases*, (pp. 385–394).

Yi, B.K., Jagadish, H.V., Faloutsos, C. (1998). Efficient retrieval of similar time sequences under time warping. In *Proceedings of the 14th international conference on data engineering*, (pp. 201–208).

Zhang, X., Liu, J., Du, Y., Lv, T. (2011). A novel clustering method on time series data. *Expert Systems with Applications*, *38*(9), 11891–11900.