A TIGHTER LOWER BOUND ESTIMATE FOR DYNAMIC TIME WARPING

Peng Yang[1], *Lei Xie*[1], *Qiao Luan*[1] *and Wei Feng*[2]

[1]Shaanxi Provincial Key Laboratory of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]School of Computer Science and Technology, Tianjin University, Tianjin, China
Email: lxie@nwpu.edu.cn

ABSTRACT

In this paper, we propose a new lower-bound estimate for speeding up dynamic time warping (DTW) on multivariate time sequences. It has several advantages as compared with the inner-product lower bound [1] recently proposed to eliminate a large number of DTW computations. First, we prove that it is $tighter$ than the inner product lower bound while the computational complexity remains comparable. Second, the inner product lower bound is specifically designed for the inner product distance while the proposed lower bound is valid for any distance measure. Third, DTW search can be further speeded up since the distance matrix is calculated in advance at the lower bound estimation stage. Spoken term detection experiments on the TIMIT corpus show that the proposed lower bound estimate is able to reduce the computational requirements for DTW-KNN search by 54% as compared with the inner-product lower bound. in black ink.

***Index Terms*—** dynamic time warping, lower-bound, spoken term detection, pattern matching

1. INTRODUCTION

Dynamic time warping (DTW) is a classical dynamic programming algorithm that searches the best alignment between two time series. Different from Euclidean distance, by allowing the compared time sequences to have different lengths, DTW provides a more reasonable similarity measure. Moreover, it has no assumption on the underlying knowledge about the sequences to be aligned. In the history of automatic speech recognition (ASR), DTW first became popular in isolated and connected word recognition and then was supplanted by hidden Markov models (HMMs), a statistical modeling framework appropriate for large vocabulary continuous speech recognition (LVCSR). However, DTW has drawn much interest recently for *unsupervised* and *low-resource* tasks, e.g., template-based speech recognition [2, 3], unsupervised speech pattern discovery [4, 5], example-based spoken term detection (STD) [6, 7] and acoustic-based spoken document segmentation [8]. Recently, Zhang *et. al.* [6] have proposed an unsupervised spoken keyword spotting approach using a segmental version of DTW. Given a speech query, a series of DTW matching is performed in windows sliding over a speech utterance.

This work is supported by the National Natural Science Foundation of China (61175018, 61100121), the Natural Science Basic Research Plan of Shaanxi Province (2011JM8009), the Fok Ying Tung Education Foundation (131059) and the Program for New Century Excellent Talents in University (NCET-11-0365).We thank Prof. Eamonn Keogh from UC Riverside and Dr. Yaodong Zhang from MIT for their great help and constructive suggestions.

Although DTW is a straightforward and intuitive sequence matching algorithm, its computational cost remains notoriously high, especially for matching in large corpora and mining trillions of time series [9]. Take DTW-based STD task for example. Given a query of length N, the computational complexity for a simple matching is $O(N^2)$. To find the best match of a given query in a speech corpus with a huge number of utterances (e.g. M), it would take $O(N^2M)$ time. For the speech pattern discovery task [4, 5], more DTW computations are required since it involves convolving a corpus of utterances against itself to find reoccurring patterns.

To speed up DTW, the concept of *lower-bound* has been introduced and various lower bound estimates have been proposed [10, 11, 1, 12]. A lower bound is able to prune off unpromising candidates and saves computation costs. Keogh *et. al.* [10] proposed an lower bound measure for comparing univariate time series, which is proved to be tighter than previous measures [13, 14]. Subsequently, Rath *et. al.* [11] extended this lower bound to multivariate time series. Recently, inner product between posteriorgram vectors [4, 1, 7] has shown superior performance on a variety of DTW-based tasks. Based on Rath's lower bound, Zhang *et. al.* [1] have proposed a lower-bound estimate specifically for the inner-product distance. For a K nearest-neighbor (KNN) spoken term detection task on the TIMIT corpus, 89% of the DTW calculations can be eliminated without affecting the detection performance. Since different lower bounds are different in tightness, Keogh has proposed to cascade different lower bounds [9].

This study is related to Rath's [11] and Zhang's [1] approaches on DTW lower bounding for multivariate time series. As we know, in order to estimate a lower bound, an auxiliary upper-bound envelope sequence needs to be firstly derived. The nature of their approaches is to calculate an *individual* upper-bound envelope for each dimension of the multivariate data vector, simply using Keogh's approach on univariate time series [10]. However, DTW distances are measured on vectors. This motivates us to develop a unified auxiliary envelope for the whole vector sequence. In this paper, we propose such a lower bound that shows several advantages as compared with Zhang's lower bound [1]. Firstly, we prove that it is $tighter$ while its computational complexity remains comparable. Secondly, Zhang's lower bound is specifically designed for the inner product distance measure while the proposed lower bound is valid for any distance measure. Finally, the distance matrix for DTW is calculated in our lower bound estimation stage and further speedup of DTW search can be achieved if the distance matrix is pre-stored. Spoken term detection experiments show that the proposed lower bound estimate is able to reduce the computational requirements for DTW-KNN search by 54% as compared with the inner-product lower bound.

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

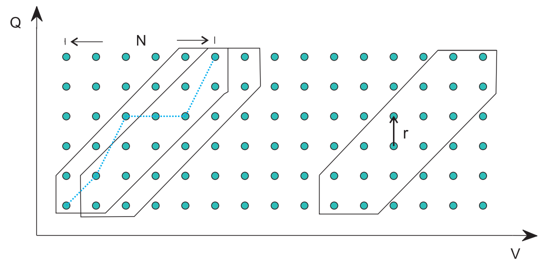

Fig. 1. Query and speech utterance matching using segmental DTW.

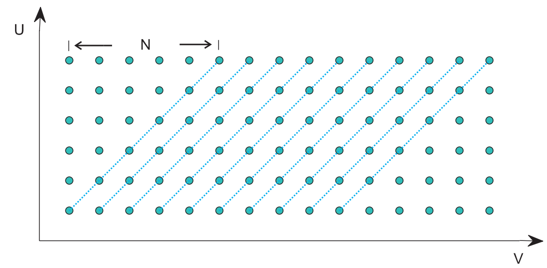

Fig. 2. The calculations for the inner product lower bound LB_{IP}.

2. BACKGROUND

2.1. Gaussian Posteriorgram

Due to the superior generalization ability across speakers, posterior probability vectors are often used as speech features [1, 12]. Gaussian posteriorgram is a feature representation of speech frames generated from a GMM [1]. A D-mixture GMM model is trained on a set of unlabeled speech frames in a corpus, $\vec{f}_1, \cdots, \vec{f}_N$. Thus a speech frame $\vec{f}_i$ can be represented by a probability vector $\vec{p}_i = \{p_i^1, \cdots, p_i^D\}$, where $p_i^j = P(g_j|\vec{f}_i)$ is a posterior probability which can be calculated for each Gaussian component $g_j \in G$, and $\sum_j p_i^j = 1 \; \forall i$.

2.2. DTW on Gaussian Posteriorgram

Given posteriorgrams of two speech sequences, $Q = \{\vec{q}_1, \cdots, \vec{q}_N\}$ and $S = \{\vec{s}_1, \cdots, \vec{s}_N\}$, where $\vec{q}_i$ and $\vec{s}_i$ are D-dimensional posterior probability vectors[1]. The distance between $\vec{q}_i$ and $\vec{s}_i$ can be defined by their inner product as $d(\vec{q}_i, \vec{s}_i) = -\log(\vec{q}_i \cdot \vec{s}_i)$ [1]. We define an alignment path $\phi = \{(\vec{q}_{\phi_q 1}, \vec{s}_{\phi_s 1}), \cdots, (\vec{q}_{\phi_q T}, \vec{s}_{\phi_s T})\}$, mapping Q to S. In order to avoid unreasonable warping paths, a global warping window constraint r is applied to ensure $|\phi_q k - \phi_s k| \leq r$. The alignment distance for path ϕ is defined as $D_\phi(Q, S) = \sum_{k=1}^{T} d(\vec{q}_{\phi_q k}, \vec{s}_{\phi_s k})$. Hence the DTW distance between Q and S is defined as

$$DTW(Q, S) = \min_{\phi} D_\phi(Q, S). \tag{1}$$

2.3. KNN-DTW based Spoken Term Detection

The spoken term detection task is to detect whether a speech utterance V (with the length of M) contains a speech query Q (with the length of N). Specifically, given a speech query example Q, we wish to find the top K nearest-neighbor (KNN) matches in speech utterances from a speech corpus. Since usually $M > N$, when comparing Q with V, we use the segmental DTW (SDTW) algorithm [6]. Specifically, a sliding window with the size equal to the length of the keyword (N) is applied to the utterance to constrain the DTW search area, as shown in Fig. 1. Note that warping constraint r is also adopted. The sliding window gradually moves (e.g., one frame forward) from the beginning to the end of the utterance, and a series of DTW matches is performed. The score for an utterance containing the keyword query corresponds to the smallest DTW score obtained in that utterance. The top K matches with the smallest DTW scores are regarded as the detected utterances that matches the input query.

[1] Without lose of generality, we use the same length N for two sequences.

2.4. The Inner-Product Lower-Bound

DTW is a matching algorithm with high computational cost. To speed up, a cheap-to-compute lower bound can be used to prune off unpromising candidates. Recently, Zhang *et. al.* [1] have proposed an inner-product lower-bound estimate for DTW on Gaussian posteriorgrams. Firstly, an upper-bound envelop sequence $U = \{\vec{u}_1, \cdots, \vec{u}_N\}$ is calculated for query Q, where $\vec{u}_i = \{u_i^1, \cdots, u_i^D\}$ and $u_i^p = \max(q_{i-r}^p, \cdots, q_{i+r}^p)$. The lower bound of DTW$(Q, S)$ can be defined as

$$LB_{IP}(Q, S) = \sum_{i=1}^{N} d(\vec{u}_i, \vec{s}_i). \tag{2}$$

Expanding the right side with the inner product definition yields $LB_{IP}(Q, S) = \sum_{i=1}^{N} -\log(\vec{u}_i \cdot \vec{s}_i)$. The proof of the lower bound property can be find in [1].

3. THE PROPOSED LOWER-BOUND

In this section, we introduce a new tighter DTW lower-bound estimate for multivariate time series, which is valid to any distance definitions.

3.1. Definition

Given two time sequences, $Q = \{\vec{q}_1, \cdots, \vec{q}_N\}$, and $S = \{\vec{s}_1, \cdots, \vec{s}_N\}$, we firstly derive a new sequence, $M = \{\vec{m}_1, \cdots, \vec{m}_N\}$, where

$$\vec{m}_i = \arg \min_{\vec{m}_i \in \{\vec{q}_{i-r}, \cdots, \vec{q}_{i+r}\}} d(\vec{m}_i, \vec{s}_i) \tag{3}$$

and r is the width of the DTW warping window. Then our lower bound can be defined as

$$LB_{YX}(Q, S) = \sum_{i=1}^{N} d(\vec{m}_i, \vec{s}_i). \tag{4}$$

3.2. Computational Complexity

Next, we will show that, for the spoken term detection task, the overall computation complexity of LB_{YX} is comparable with LB_{IP}. Fig. 2 shows the lower-bound calculations for LB_{IP}. A dot denotes the distance between a frame in the upper-bound envelop sequence U of the query keyword Q and a frame in the speech utterance V. According to the sliding strategy in the SDTW algorithm, a lower-bound is calculated for each blue dotted line in Fig. 2. Specifically, the sum of the distances of the dots on the blue dotted line is used as the lower-bound, as defined in Eq. (2). Therefore, to search for a query in a speech utterance, the overall computation complexity for the LB_{IP} estimates is $O(MN)$. For the calculation of LB_{YX}, we can refer to Fig. 1. In each hexagon area (warping window constraint

$r = 1$), we seek for the dot with the smallest distance in each column, as shown in Eq. (3). According to Eq. (4), we sum the distances of these dots on the blue line and obtain the lower-bound. To search for a query in a speech utterance, sliding the hexagon from the left to the right, the overall computation complexity for the LB_{YX} estimates is also $O(MN)$. Please note that it might take a little bit longer to calculate LB_{YX} since the dots involved in LB_{YX} calculation is a little bit more than those in LB_{IP}. This can be seen when comparing Fig. 1 with Fig. 2.

3.3. Proof: Lower Bound Property

To prove the lower-bounding property, i.e.,

$$LB_{YX}(Q, S) \leq DTW(Q, S), \tag{5}$$

we follow the thoughts in [10, 11]. Assuming the alignment path $\hat{\phi} = \{(\vec{q}_{\hat{\phi}_q 1}, \vec{s}_{\hat{\phi}_s 1}), \cdots, (\vec{q}_{\hat{\phi}_q T}, \vec{s}_{\hat{\phi}_s T})\}$, corresponds to the minimum alignment distance, where T denotes the length of the path, we have $DTW(Q, S) = \sum_{k=1}^{T} d(\vec{q}_{\hat{\phi}_q k}, \vec{s}_{\hat{\phi}_s k})$. By expanding both terms in Eq.(5), we need to show

$$\sum_{i=1}^{N} d(\vec{m}_i, \vec{s}_i) \leq \sum_{k=1}^{T} d(\vec{q}_{\hat{\phi}_q k}, \vec{s}_{\hat{\phi}_s k}) \tag{6}$$

Since the summation terms on both sides of Eq.(6) is positive, we just need to prove the inequality in Eq.(6) by showing that, for each term on the left side, there exists an equal or greater term on the right side. To this end, we split the the right side into two parts

$$\sum_{i=1}^{N} d(\vec{m}_i, \vec{s}_i) \leq \sum_{k \in \text{MA}} d(\vec{q}_{\hat{\phi}_q k}, \vec{s}_{\hat{\phi}_s k}) + \sum_{k \in \text{UM}} d(\vec{q}_{\hat{\phi}_q k}, \vec{s}_{\hat{\phi}_s k}), \tag{7}$$

where MA denotes a matched set containing exactly N warping pairs, and UM denotes to an unmatched set including all remaining pairs. We form the two sets as follows. For the i^{th} term on the left side, a warping pair $(\hat{\phi}_q k, \hat{\phi}_s k)$ from the right side is chosen into MA if $\hat{\phi}_s k = i$. If there are multiple warping pairs on the right side with $\hat{\phi}_s k = i$, we just select the pair with smallest $\hat{\phi}_s k = i$. Since $T \geq n$, there are always enough pairs can be selected into MA. By following this rule we ensure that the size of MA is exactly N so that each term on the left side is matched exactly once by an element in the MA on the right side, i.e., one-to-one match. Hence if we are able to show

$$\sum_{i=1}^{N} d(\vec{m}_i, \vec{s}_i) \leq \sum_{k \in \text{MA}} d(\vec{q}_{\hat{\phi}_q k}, \vec{s}_{\hat{\phi}_s k}), \tag{8}$$

Eq. (6) can be obviously proved.

Consider an individual warping pair in MA, $(\hat{\phi}_q k, \hat{\phi}_s k)$, which corresponds to the i^{th} term on the left side. We need to show

$$d(\vec{m}_i, \vec{s}_i) \leq d(\vec{q}_{\hat{\phi}_q k}, \vec{s}_{\hat{\phi}_s k}) \tag{9}$$

According to the DTW constraint window, $i - r \leq \hat{\phi}_q k \leq i + r$, we have $\vec{q}_{\hat{\phi}_q k} \in \{\vec{q}_{i-r}, \cdots, \vec{q}_{i+r}\}$. Due to Eq. (3), $d(\vec{m}_i, \vec{s}_i)$ is the smallest distance within the warping constraint window. Obviously Eq. (9) is proved and from bottom to up, Eq. (5) is proved. Please note that this lower bound property is valid for any distance measure as compared to LB_{IP} [1] whose lower bound property is proved under the inner-product distance measure.

3.4. Proof: Tighter than LB_{IP}

Here, we show that the new lower-bound estimate is tighter than the inner-product one, i.e., $LB_{YX} \geq LB_{IP}$. Take the inner products in,

$$\sum_{i=1}^{n} -\log(\vec{m}_i \cdot \vec{s}_i) \geq \sum_{i=1}^{n} -\log(\vec{u}_i \cdot \vec{s}_i). \tag{10}$$

If we are able to show every sum term on the left-hand side is greater than or equal to the corresponding term on the right-hand side, i.e.,

$$-\log(\vec{m}_i \cdot \vec{s}_i) \geq -\log(\vec{u}_i \cdot \vec{s}_i), \tag{11}$$

Eq. (10) is proved. By eliminating the log and the minus sign, we need to show

$$\vec{m}_i \cdot \vec{s}_i \leq \vec{u}_i \cdot \vec{s}_i. \tag{12}$$

Expanding the both sides of Eq.(12) yields

$$\sum_{j=1}^{D} m_i^j \times s_i^j \leq \sum_{j=1}^{D} u_i^j \times s_i^j, \tag{13}$$

which can also be proved by showing that every sum term on the left is less than or equal to the corresponding term on the right,

$$m_i^j \times s_i^j \leq u_i^j \times s_i^j. \tag{14}$$

Since $\vec{m}_i = \arg\min_{\vec{m}_i \in \{\vec{q}_{i-r}, \cdots, \vec{q}_{i+r}\}} d(\vec{m}_i, \vec{s}_i)$, we have $m_i^j \in \{q_{i-r}^j, \cdots, q_{i+r}^j\}$. By definition in [1], $u_i^j = \max(q_{i-r}^j, \cdots, q_{i+r}^j)$. Thus $m_i^j \leq u_i^j$ holds. According to the definition of the Gaussian Posteriorgram, s_i^j, u_i^j and m_i^j are all positive. Therefore $m_i^j \times s_i^j \leq u_i^j \times s_i^j$ and from bottom to up, Eq. (10) is proved.

We show that our lower bound estimate is tighter than the inner-product one [1]:

$$DTW(Q, S) \geq LB_{YX}(Q, S) \geq LB_{IP}(Q, S). \tag{15}$$

Theoretically a tighter lower bound can prune more DTW calculations and the KNN search task can be speeded up. In Section 4, experiments will confirm this.

3.5. Further KNN-DTW Speedup

As we know, to match two sequences using DTW, an $N \times N$ distance matrix needs to be calculated. As we described in Section 3.1, this distance matrix has been already calculated during our lower-bound estimate stage. Therefore, if this matrix is stored in advance, the subsequent KNN-DTW search can just work on this matrix and we do not need to calculate it again. By this way, further speedup can be achieved. At the same time, the space complexity is increased to $O(MN)$, where N is the keyword's length, and M is the whole length of the utterances in the dataset. DTW matching using LB_{IP} does not has this advantage of speedup. Actually, in order to calculate LB_{IP}, an $N \times N$ distance matrix between U and S has to be calculated. During DTW matching, part of another $N \times N$ distance matrix between Q and S still needs to be calculated, although some of the distance calculations are eliminated according to the lower bound.

4. EXPERIMENTS

4.1. Experimental Setup

The speedup ability of the two lower-bound estimates (LB_{YX} and LB_{IP}), is compared on an example-based spoken term detection task as described in Section 2.3. The distance matrix pre-storage strategy introduced in Section 3.5 is also tested and we name this approach as LB_{YXS}. We follows the experimental setup in [12]. The TIMIT corpus is divided into a training set with 3,696 utterances and a test set with 944 utterances. Each utterance is then converted into a series of 13-dimension MFCC vectors (frame length:25ms, shift rate: 10ms). A 50 component GMM is then trained in an unsupervised manner using the training set. As a result, every speech frame in the training and test sets is represented by a 50-dimension GMM posteriorgram vector.

A set of 10 keywords is randomly selected and one example of each keyword is extracted from the training set as the input query. A stopword list is used to avoid the frequently used words from being selected as keywords. According to Section 2.3, the spoken term detection task is to find the K best matching utterances in the test set containing the query keyword. For the experiments on the three lower-bound estimates, we use the same keyword list and the same DTW-KNN search framework proposed in [1].

4.2. Experimental Results

From Section 3.3, we can see that the proposed lower-bound estimate (LB_{YX}) is admissible. The lowest equal error rate (EER) achieved by the LB_{YX} approach is 14.58% when $K = 143$ and $r = 5$. This EER is almost the same with the result achieved by LP_{IP} [1]. We notice that the K and r that achieve the lowest EER are different for the two lower bounds. This is probably because our experimental configuration (e.g., feature extraction parameters, query examples, etc.) is slightly different from [1].

Fig. 3 illustrates the average inner product save ratio against different K nearest neighbors for different r. The ratio is defined as the percentage of total inner product calculations saved by LB_{YX} comparing with LB_{IP}. As seen in the figure, in most cases, the proposed lower bound achieves apparent computational savings. The highest computing saving is 12.2%, which is achieved when $K = 150$ and $r = 9$. When the lowest EER is achieved ($K = 143$, $r = 5$), the inner product save rate is 8%. We also notice that when r and K are small, LB_{YX} shows slightly worse performance. This can be explained as follows. The tightness of lower bounds and the pruning power essentially depend on the size of the warping window r, and the smaller the area of allowed warping, the more we can take advantage of pruning [12]. Thus when r is small , the two lower-bound estimates have comparable tightness and pruning power. Although calculating LB_{YX} and LB_{IP} is in the same order of magnitude of computation ($O(MN)$), LB_{YX} may need a little bit more calculations, as explained in Section 3.2.

Fig. 3 shows the computational savings caused only by the *tightness* of the new lower bound. If the pre-storage approach is considered, more calculations can be saved, as shown in Fig. 4. Here, the save ratio is defined as the percentage of total inner product calculations saved by LB_{YXS} comparing with LB_{IP}. We can see that more calculations can be eliminated when using the pre-storage approach. Specifically, when $K = 143$ and $r = 5$, at which setting the minimum EER is achieved, the LB_{YXS} approach can save 54% inner-product calculations as compared to LB_{IP}. We also notice that the inner product save ratio increases with r. This behavior can be explained as follows. A bigger r leads to less tighter lower-bound estimate, and thus more DTW searches need to be done, as can be obviously seen in [1]. As a result, the increased DTW searches cause more inner-product calculations. The LB_{YXS} method stores the distance matrix in advance, and thus the increased DTW search doesn't bring increased inner-product calculations. Hence the save ratio becomes bigger as r increases.

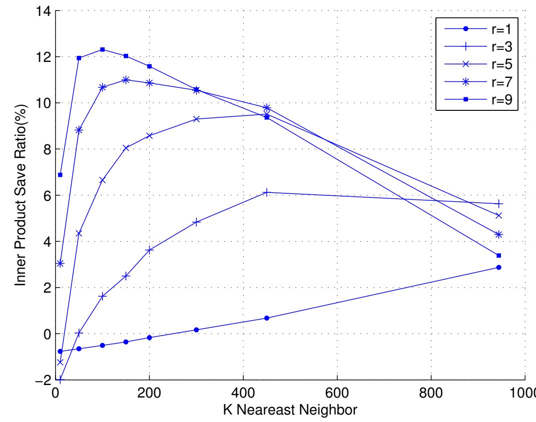

Fig. 3. The average inner product calculations save ratio against different K nearest neighbor; $Ratio = (Inner\_Prod(LB_{IP}) - Inner\_Prod(LB_{YX}))/Inner\_Prod(LB_{IP})$.

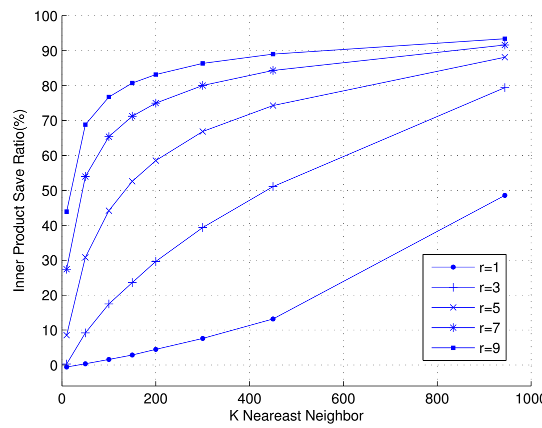

Fig. 4. The average inner product calculations save ratio against different K nearest neighbor; $Ratio = (Inner\_Prod(LB_{IP}) - Inner\_Prod(LB_{YXS}))/Inner\_Prod(LB_{IP})$.

In terms of the empirical computation time, the inner product lower bound approach (LB_{IP}) takes 48.4s, while the new lower bound (LB_{XYS}) take only 27.1s, when searching matches for the 10 keywords in the test set on a single desktop CPU on average.

5. CONCLUSIONS

In this paper, we have proposed a tighter lower bound estimate for DTW on multivariate time series. Experiments on a DTW-KNN spoken term detection task have shown that the lower bound is superior to the inner product lower bound in speeding up the DTW search. Since our lower bound is valid to any distance measure, we plan to implement our lower bound to other distance measures. To further reduce the computation burden from lower bound estimate, Zhang *et. al.* have recently proposed a cheap-to-compute lower-bound based on piecewise aggregate approximation (PAA) [12]. PAA can be viewed as a down-sampling approach which can make a short but representative abstraction for a long time series. It reduces the calculation of the lower-bound estimate, leading to a less tighter lower-bound, but produces an overall KNN search speedup. In the future, we plan to develop a similar down-sampling approach for our lower bound and compare it with the PAA approach.

6. REFERENCES

[1] Y. Zhang and J. Glass, "An inner-product lower-bound estimate for dynamic time warping," in *Proc. ICASSP*, 2011.

[2] L. Deng and H. Strik, "Structure-based and template-based automatic speech recognition: Comparing parametric and non-parametric approaches," in *Proc. Interspeech*, 2007.

[3] M. D. Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. V. Compernolle, "Template-based continuous speech recognition," *IEEE Trans. ASLP*, vol. 15, no. 4, 2007.

[4] A. Park and J. Glass, "Unsupervised pattern discovery in speech," *IEEE Trans. ASLP*, vol. 16, no. 1, 2008.

[5] L. Xie, Y. Xu, L. Zheng, Q. Huang, and B. Li, "Speech pattern discovery using audio-visual fusion and canonical correlation analysis," in *Proc. Interspeech*, 2012.

[6] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *Proc. ASRU*, 2009.

[7] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. ICASSP*, 2012.

[8] L. Zheng, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Acoustic texttiling for story segmentation of spoken documents," in *Proc. ICASSP*, 2012, pp. 5121–5124.

[9] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. SIGKDD*, 2012.

[10] E. Keogh, "Exact indexing of dynamic time warping," in *Proc. VLDB*, 2002.

[11] T. M. Rath and R. Manmatha, "Lower-bounding of dynamic time warping distances for multivariate time series," Tech.Rep.MM-40, University of Massachusetts, 2002.

[12] Y. Zhang and J. Glass, "A piecewise aggregate approximation lower-bound estimate for posteriorgram-based dynamic time warping," in *Proc. Interspeech*, 2011.

[13] B.-K. Yi, H. V. Jagadish, and C. Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *Proc. ICDE*, 1998.

[14] S.-W. Kim, S. Park, and W. W. Chu, "An index- based approach for similarity search supporting time warping in large sequence databases," in *Proc. ICDE*, 2001.