# A Piecewise Aggregate Approximation Lower-Bound Estimate for Posteriorgram-based Dynamic Time Warping

*Yaodong Zhang and James Glass*

MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, Massachusetts 02139, USA
{ydzhang,glass}@csail.mit.edu

## Abstract

In this paper, we propose a novel lower-bound estimate for dynamic time warping (DTW) methods that use an inner product distance on multi-dimensional posterior probability vectors known as posteriorgrams. Compared to our previous work, the new lower-bound estimate uses piecewise aggregate approximation (PAA) to reduce the time required for calculating the lower-bound estimate. We describe the PAA lower-bound construction process and prove that it can be efficiently used in an admissible $K$ nearest neighbor (KNN) search. The amount of computational savings is quantified by a set of unsupervised spoken keyword spotting experiments. The results show that the newly proposed PAA lower-bound is able to speed up DTW-KNN search by 28% without affecting the keyword spotting performance.

**Index Terms**: dynamic time warping, lower-bound, posteriorgram

## 1. Introduction

Dynamic Time Warping (DTW) is a broadly explored technique for aligning two time indexed patterns. The key advantage of DTW is that it assumes no underlying knowledge about the patterns to be aligned, and can provide a quantitative solution for measuring similarity [1]. In terms of computational complexity for two patterns of length $M$, a DTW match takes $O(M^2)$ to output the best alignment between these two patterns. However, given a query pattern, if DTW is used to search similar patterns in a large speech corpus, the squared time complexity would become a very large computational burden. For example, to find the best match of a given speech pattern of length $M$ in a large speech corpus with $N$ entries, DTW would take $O(M^2 \cdot N)$ time where $N$ could be very large.

To address this problem, several lower-bound estimation algorithms have been proposed for DTW-KNN search in large corpora [2, 3, 4]. There are two basic ideas behind lower-bounded DTW-KNN search. First, a lower-bound distance that underestimates the actual DTW distance is computed between the query pattern and all candidate patterns in the corpus. Second, to search for the nearest neighbor of the query pattern, the algorithm starts from the candidate pattern with the smallest lower-bound estimate and calculates the actual DTW distance as the current best match. It is clear that based on the definition of the lower-bound, any candidate patterns with lower-bound estimates greater than the current best match can be pruned away. The nearest neighbor search is complete when all remaining candidate patterns have lower-bound estimates greater than the current best match.

Inspired by the lower-bound DTW-KNN search idea, in our recent work we proposed a lower-bound estimate for DTW-KNN search on posteriorgrams using an inner product distance [5]. For a keyword spotting task on the TIMIT corpus, we observed that 89% of the DTW calculations can be eliminated without affecting the keyword spotting performance. The time consumed for the lower-bound estimate is $O(M)$ for a keyword query with $M$ frames. For a database with $N$ entries however, calculating the lower-bound estimate for a keyword query and every candidate speech segment in the database would take $O(MN)$, which could still be a considerable burden when $N$ is very large (e.g., $N > 10^7$).

To address this disadvantage, in this paper, we propose an improved lower-bound estimate using piecewise aggregate approximation (PAA). PAA can be viewed as a down-sampling approach which can make a short but representative abstraction for a long time series. When comparing two time series, using their corresponding PAA representation saves computation time in exchange for a slightly weaker lower-bound estimate. Prior research with PAA has focused on using the Euclidean distance as the similarity metric between one dimensional time series [2, 6]. Our recent work has been using DTW on posteriorgrams (a series of multi-dimensional posterior probability vectors) based on the inner product distance metric [5, 7]. If we therefore hope to leverage the PAA concept for reducing lower-bound calculations, we need to develop a new PAA lower-bound estimate for posteriorgrams. Therefore, we describe a PAA lower-bound estimate approach, and prove that it is admissible for DTW-KNN search. Using a PAA representation of posteriorgrams might lead to a weaker lower-bound estimate which would increase the necessary DTW calculations in a KNN search. In our reported keyword spotting experiments on the TIMIT corpus we consider the total calculation needed for both the lower-bound estimate and DTW-KNN search. The results showed that the proposed PAA lower-bound estimate reduced the computational requirements for DTW-KNN search by 28% compared with our previous best lower-bound estimate approach [5].

## 2. Background

In this section we first briefly review the concept of the Gaussian posteriorgram representation and the associated lower-bound estimate for DTW-KNN search.

### 2.1. Gaussian Posteriorgram

Given speech frames $f_1, f_2, \cdots, f_n$, the Gaussian posteriorgram for each speech frame $f_i$ is a collection of posterior probabilities $P(g_j|f_i)$. $g_j \in G$ is a mixture from a Gaussian mixture model $G$ which is trained on a set of unlabeled speech frames. If $G$ consists of $D$ mixtures, each speech frames can be represented by a $D$-dimensional vector $\vec{p}_i = \{p_i^1, p_i^2, \cdots, p_i^D\}$, where $\sum_j p_i^j = 1, \forall i$.

### 2.2. Lower-Bound Estimate for DTW on Posteriorgram

Consider two Gaussian posteriorgrams for a speech query $Q = \{\vec{q}_1, \ldots, \vec{q}_M\}$ and a speech segment $S = \{\vec{s}_1, \ldots, \vec{s}_N\}$, the local distance between $\vec{q}_i$ and $\vec{s}_j$ is defined by their inner product distance as $d(\vec{q}_i, \vec{s}_j) = -\log(\vec{q}_i \cdot \vec{s}_j)$. Given an alignment warp $\phi = (\phi_q, \phi_s)$ of length $K_\phi$, the corresponding alignment score $A_\phi(Q, S)$ is obtained from the sum of the local distances

$$A_\phi(Q,S) = \sum_{k=1}^{K_\phi} d(\vec{q}_{\phi_q(k)}, \vec{s}_{\phi_s(k)})$$

where $1 \le \phi_q(k) \le M$ and $1 \le \phi_s(k) \le N$. The overall best alignment score $\text{DTW}(Q,S) = \min_\phi A_\phi(Q,S)$. A global path constraint $r$ is usually applied to ensure that the warp will be within $r$ frames of each other along the entire alignment.

Given a query posteriorgram, if we wish to find the top $K$ nearest neighbor matches in a speech corpus, the traditional DTW search method needs to go through all the candidate segments and rank the DTW distance between the query posteriorgram and every candidate segment. However, with the help of a lower-bound estimate of the DTW distance, the KNN search can terminate if all the remaining candidates have lower-bound estimates greater than the $K^{th}$ best match. The construction of the lower-bound estimate is as follows. Given two posteriorgrams $Q$ and $S$, an upper-bound envelope sequence $U$ is calculated on $Q$, where $U = \{\vec{u}_1, \cdots, \vec{u}_M\}$, $\vec{u}_i = \{u_i^1, \cdots, u_i^D\}$ and $u_i^p = \max(q_{i-r}^p, \cdots, q_{i+r}^p)$. $U$ can be viewed as a sliding-maximum on $Q$ with window size $r$ [5]. Note that $r$ is the DTW global path constraint mentioned earlier. The lower-bound estimate of $\text{DTW}(Q,S)$ can be defined as

$$L(Q,S) = \sum_{i=1}^{l} d(\vec{u}_i, \vec{s}_i) \quad (1)$$

where $l = \min(M,N)$. The time required for computing $L(Q,S)$ is only $O(l)$. In [5], we proved that

$$L(Q,S) \le DTW(Q,S)$$

which guarantees no false dismissals for DTW-KNN search.

## 3. Piecewise Aggregate Approximation

Although using the lower-bounded DTW-KNN search can save a considerable amount of DTW calculation, the computation of the lower-bound estimate itself is still time consuming. To further improve the efficiency, inspired by [2, 6], we apply the concept of piecewise aggregate approximation (PAA) to reduce the length of posteriorgrams into $B$ blocks, and estimate a slightly weaker lower-bound in exchange for a faster lower-bound calculation.

### 3.1. Definition

Given two posteriorgrams $Q$ and $S$, without loss of generality, we assume they have the same length $M = N$. Define two approximation posteriorgrams $\hat{U} = \{\hat{U}_1, \cdots, \hat{U}_B\}$ and $\hat{S} = \{\hat{S}_1, \cdots, \hat{S}_B\}$. $\hat{U}_i$ denotes the $i^{th}$ block of the approximated upper-bound envelope $U$, and is defined as $\hat{U}_i = \{\hat{u}_i^1, \cdots, \hat{u}_i^D\}$ where each dimension

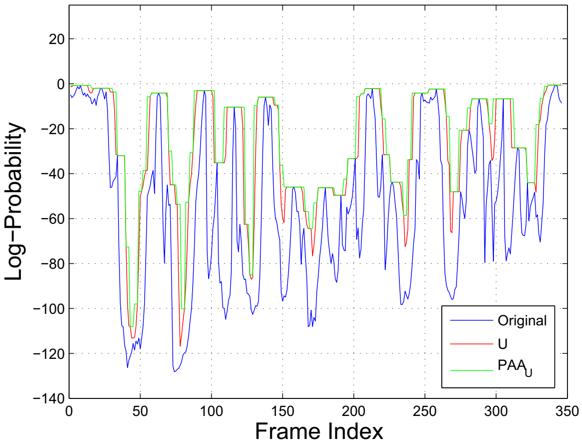

Figure 1: *Example of a one-dimensional PAA sequence (3 frames per block) (green), an upper-bound envelope sequence (red) and an original posteriorgram (blue) for r = 5.*

$$\hat{u}_i^p = \max\left(u^p_{\frac{M}{B}(i-1)+1}, \cdots, u^p_{\frac{M}{B}i}\right) \quad (2)$$

$\hat{S}_i$ denotes the $i^{th}$ block of the approximated $S$ and can be defined as $\hat{S}_i = \{\hat{s}_i^1, \cdots, \hat{s}_i^D\}$ where each dimension

$$\hat{s}_i^p = \frac{B}{M} \sum_{j=\frac{M}{B}(i-1)+1}^{\frac{M}{B}i} s_j^p \quad (3)$$

Note that if $M$ is not divisible by $B$, $\frac{M}{B}$ is floored and the remaining frames form an additional block. It is clear that the PAA block reduction process is similar with a down-sampling process. For a speech query, for each dimension the maximum value of the frames within a block is used to represent the block, while for a speech segment, the average value of the frames within a block is used. Figure 1 demonstrates an example of the approximated $\hat{Q}$ and the upper-bound envelope $U$ on one dimension of a posteriorgram.

Using $\hat{U}$ and $\hat{S}$, the PAA lower-bound estimate for DTW on posteriorgrams can be defined as

$$\text{PAA}(Q,S) = \sum_{i=1}^{B} \frac{M}{B} \cdot d(\hat{U}_i, \hat{S}_i) \quad (4)$$

where $d(\cdot)$ is the inner product function.

### 3.2. Proof

To prove $\text{PAA}(Q,S) \le L(Q,S)$, without loss of generality, we first assume that $B = 1$ which indicates the entire posteriorgram sequence is considered as one block. If under this assumption the inequality holds, it is clear that the same proof can be applied to each block when $B \ge 1$. (Note that if $B = M$ then $\text{PAA}(Q,S) = L(Q,S)$.)

Since $B = 1$, Eq. 4 can be simplified as

$$\text{PAA}(Q,S) = M \cdot \left(-\log \sum_{p=1}^{D} \hat{u}_1^p \cdot \hat{s}_1^p\right)$$

According to the definition of the original lower-bound estimate in Eq. 1, the inequality becomes

$$M \cdot \left( -\log \sum_{p=1}^{D} \hat{u}_1^p \cdot \hat{s}_1^p \right) \leq \mathrm{L}(Q, S) = \sum_{i=1}^{M} -\log \sum_{p=1}^{D} u_i^p \cdot s_i^p$$

After absorbing the summation into the log and negating both sides, the inequality becomes

$$\log \left( \sum_{p=1}^{D} \hat{u}_1^p \cdot \hat{s}_1^p \right)^M \geq \log \prod_{i=1}^{M} \sum_{p=1}^{D} u_i^p \cdot s_i^p$$

which is equivalent to prove

$$\left( \sum_{p=1}^{D} \hat{u}_1^p \cdot \hat{s}_1^p \right)^M \geq \prod_{i=1}^{M} \sum_{p=1}^{D} u_i^p \cdot s_i^p \tag{5}$$

Note that since $B = 1$, according to the definition of the block reduction process in Eq. 2 and Eq. 3, it is clear that

$$\hat{u}_1^p = \max\left(u_1^p, u_2^p, \cdots, u_M^p\right)$$
$$\hat{s}_1^p = \frac{1}{M} \sum_{i=1}^{M} s_i^p$$

Therefore, the left hand side of Eq. 5 can be written as

$$\begin{aligned} \left( \sum_{p=1}^{D} \hat{u}_1^p \cdot \hat{s}_1^p \right)^M &= \left( \frac{1}{M} \sum_{p=1}^{D} \sum_{i=1}^{M} \hat{u}_1^p \cdot s_i^p \right)^M \\ &\geq \left( \frac{1}{M} \sum_{p=1}^{D} \sum_{i=1}^{M} u_i^p \cdot s_i^p \right)^M \end{aligned} \tag{6}$$

where $\hat{u}_1^p \geq u_i^p, \forall i \in [1, M]$ based on Eq. 2. Interchanging the summation in Eq. 6, the inequality we need to prove becomes

$$\left( \frac{1}{M} \sum_{i=1}^{M} \sum_{p=1}^{D} u_i^p \cdot s_i^p \right)^M \geq \prod_{i=1}^{M} \sum_{p=1}^{D} u_i^p \cdot s_i^p \tag{7}$$

Let $a_i = \sum_{p=1}^{D} u_i^p \cdot s_i^p$, the inequality becomes

$$\left( \frac{1}{M} \sum_{i=1}^{M} a_i \right)^M \geq \prod_{i=1}^{M} a_i$$

Since it is clear that $a_i \geq 0$, the arithmetic mean is always greater than or equal to the geometric mean. Combining with the proof in [5], the following inequality holds

$$\mathrm{PAA}(Q, S) \leq \mathrm{L}(Q, S) \leq \mathrm{DTW}(Q, S) \tag{8}$$

which indicates the PAA lower-bound estimate is admissible to DTW-KNN search.

Since the sum of posterior probabilities in a posteriorgram should be one, in order to avoid trivialness we should prove (as in [5]) that the approximated posteriorgram has the property that

$$\left( \sum_{p=1}^{D} \hat{u}_1^p \cdot \hat{s}_1^p \right)^M \leq 1$$

From Eq. 6 and Eq. 7, it is clear that

$$\sum_{p=1}^{D} \hat{u}_1^p \cdot \hat{s}_1^p = \frac{1}{M} \sum_{i=1}^{M} \sum_{p=1}^{D} \hat{u}_1^p \cdot s_i^p$$

Let $\hat{u}_{max} = \max(\hat{u}_1^1, \hat{u}_1^2, \cdots, \hat{u}_1^D)$. We have

$$\begin{aligned} \frac{1}{M} \sum_{i=1}^{M} \sum_{p=1}^{D} \hat{u}_1^p \cdot s_i^p \leq \frac{\hat{u}_{max}}{M} \cdot \sum_{i=1}^{M} \sum_{p=1}^{D} s_i^p = \frac{\hat{u}_{max}}{M} \cdot \sum_{i=1}^{M} \cdot 1 \\ = \frac{\hat{u}_{max}}{M} \cdot M \cdot 1 = \hat{u}_{max} \leq 1 \end{aligned}$$

where $\sum_{p=1}^{D} s_i^p = 1$ based on the posteriorgram definition.

# 4. Evaluation

Since the proposed PAA lower-bound estimate is a direct extension to our previous work, the evaluation and comparison was performed on the same task as in [7].

## 4.1. The Unsupervised Keyword Spotting Experiment

The unsupervised keyword spotting experiment was performed on the TIMIT corpus, including a training set of 3,696 utterances and a test set of 944 utterances. Each utterance was segmented into a series of 25 ms frames with a 10 ms analysis rate; each frame was represented by 13 MFCC features. A GMM with 50 Gaussian components was trained on the frames in the training set. Then, frames from both the training set and the test set were decoded by the GMM, producing a 50-dimension posteriorgram vector for each frame.

Ten keywords were randomly selected and one example of each keyword was extracted from the training set. A stop list is applied to prevent frequently used words from being selected as a keyword. The keyword spotting task was to find the $K$ best matching utterances from the test set containing each keyword. In [7], we proposed a DTW-KNN search framework using lower-bound estimate in Eq. 1. Specifically, the KNN search terminates when $K$ utterances are found and the lower-bound estimates of all remaining utterances are greater than the $K^{th}$ best DTW score. The same DTW-KNN search framework was applied except that the lower-bound estimate was replaced by the newly proposed PAA lower-bound estimate in Eq. 4.

In order to analyze computational costs, one inner product calculation of two posteriorgrams is viewed as an atomic operation. For example, if the speech query and the test speech segment have length $M$, the original lower-bound estimate would require $M$ inner products while the proposed PAA lower-bound estimate requires $B$ inner products, where $B \leq M$ is the number of blocks after applying PAA. When searching for a keyword, the sum of the inner products needed in the lower-bound estimate and the following DTW calculation is considered to be the total amount of required computation. Note that the computational overhead needed for the PAA block reduction in Eq. 2 and Eq. 3 is small. Specifically, the approximated upper-bound envelope is calculated only once for each speech query, while for speech segments in the database, the average value of frames within a block can be pre-calculated and stored.

## 4.2. Experimental Results

Since the PAA lower-bound estimate is admissible (Eq. 8), the keyword spotting accuracy is the same as the previously reported result which was 14.6% equal error rate (EER) [7].

Figure 2 illustrates why the proposed PAA lower-bound estimate can speed up the overall calculation compared to the original lower-bound method. The green dotted curve (LB) represents the actual inner product calculations needed for lower-bound estimate only. Each solid curve (DTW) represents the ac-

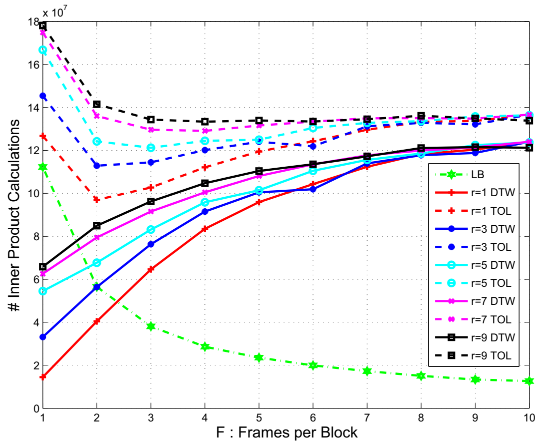

Figure 2: *Actual inner product calculation against different number of frames per block.*

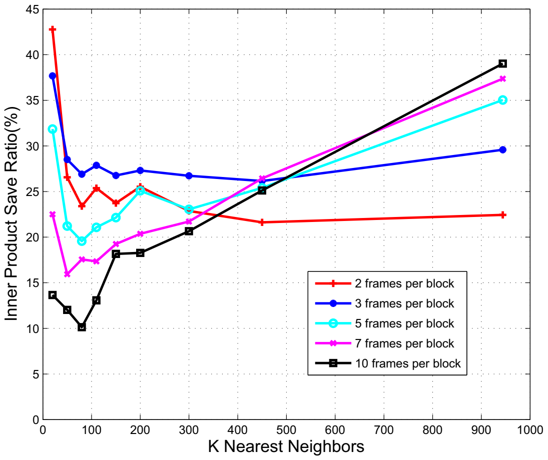

Figure 3: *Average inner product calculation save ratio against different $K$ nearest neighbors.*

tual inner product calculations needed for the DTW calculation with different $r$. $r$ is the global path constraint in DTW. Each dashed curve (TOL) is the sum of the solid curve (in the same color) and the green dotted curve, which indicates the total inner product calculations needed. Note that, as mentioned earlier, when there is only one frame per block, the PAA lower-bound estimate degrades to the original lower-bound estimate. For example, when $F = 1, r = 5$, the original approach requires $1.12 \times 10^8$ inner product calculations as well as $5.45 \times 10^7$ for the DTW calculations. In this case, the time consumed on the lower-bound estimate is more than the time consumed on the DTW calculations. However, if we increase the number of frames per block, the number of inner product calculations required for the lower-bound estimates decreases. At the same time, since the lower-bound estimates become weaker, the number of inner product calculations required for the DTW increases. Considering the total inner product calculations, it can be seen that a large amount of inner product calculations can be saved compared with the original approach ($F = 1$). Since according to [7], the minimum EER was achieved when $r = 5$, the PAA lower-bound estimate for this $r$ value can save 28% of the inner product calculations when $F = 3$.

Figure 3 compares the average inner product save ratio against different $K$ nearest neighborhoods when $r = 5$. The inner product save ratio is defined as the percentage of total inner product calculations saved comparing with the original lower-bound estimate. As seen in the figure, for this task both small and large values of $K$ achieve greater overall computational savings compared to values of $K$ between 100 and 450. We believe the reason for this behavior is because when $K$ is small, searching $K$ best matches is a highly competitive process. A slightly weaker lower-bound estimate will increase DTW calculations dramatically so that the inner product save ratio is dominated by the inner product calculations needed on the DTW side. As a result, the save ratio is inversely proportional to the number of frames per block because having more frames in a block results in a greater underestimation of the lower-bound. In contrast, for large $K$ the KNN searches almost all speech segments. The inner product save ratio largely depends on the number of inner product calculations in the lower-bound estimate. Thus, the PAA lower-bound estimates with large block sizes achieve greater overall savings.

In terms of computation time, the original lower-bound approach takes 120 seconds on a single desktop CPU on average, while the PAA lower-bound method needs 87 seconds. Since the TIMIT test corpus contains 48 minutes of speech, each keyword search takes approximately 10 seconds/query/corpus hour/CPU, compared with 14 seconds/query/corpus hour/CPU achieved with the original lower-bound estimate.

## 5. Conclusion

In this paper, we present a novel admissible lower-bound estimate for DTW based on the inner product distance on multidimensional posteriorgrams. The new lower-bound estimate uses piecewise aggregate approximation (PAA) to reduce the computational requirements for the estimate. Using PAA leads to a weaker lower-bound estimate, which in turn increases the number of DTW calculations required in a KNN search. When considering the total calculation needed for both lower-bound estimate and DTW-KNN search, TIMIT keyword spotting experiments indicate that the proposed PAA lower-bound estimate is able to speed up the overall DTW-KNN search by 28%.

Since the lower bound calculation can be easily parallelized, in future work, we would like to examine other computing architectures such as GPU computing to further speed up the entire algorithm.

## 6. References

[1] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Prentice-Hall, Inc., 1993.

[2] E. Keogh, "Exact indexing of dynamic time warping," in *Proc. VLDB*, 2002, pp. 406–417.

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

[4] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," in *Proc. SIGKDD*, 2003, pp. 216–225.

[5] Y. Zhang and J. Glass, "An inner-product lower-bound estimate for dynamic time warping," in *Proc. ICASSP*, 2011, pp. 5660–5663.

[6] B. Yi and C. Faloutsos, "Fast time sequence indexing for arbitrary $L_p$ forms," in *Proc. VLDB*, 2000, pp. 385–394.

[7] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU*, 2009, pp. 398–403.