AN INNER-PRODUCT LOWER-BOUND ESTIMATE FOR DYNAMIC TIME WARPING

Yaodong Zhang and James R. Glass

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

ABSTRACT

In this paper, we present a lower-bound estimate for dynamic time warping (DTW) on time series consisting of multi-dimensional posterior probability vectors known as posteriorgrams. We develop a lower-bound estimate based on the inner-product distance that has been found to be an effective metric for computing similarities between posteriorgrams. In addition to deriving the lower-bound estimate, we show how it can be efficiently used in an admissible K nearest neighbor (KNN) search for spotting matching sequences. We quantify the amount of computational savings achieved by performing a set of unsupervised spoken keyword spotting experiments using Gaussian mixture model posteriorgrams. In these experiments the proposed lower-bound estimate eliminates 89% of the DTW previously required calculations without affecting overall keyword detection performance.

Index Terms— dynamic time warping, posteriorgram

1. INTRODUCTION

Dynamic Time Warping (DTW) is a well-known dynamic programming technique for finding the best alignment between two time series patterns. DTW became popular in the automatic speech recognition (ASR) community from the late 1970's to mid 1980's, and was used for both isolated and connected-word recognition with spectrally-based representations such as LPC [1]. DTW allowed for minor local time variations between two speech patterns which made it a simple and efficient search mechanism. Over time, DTW-based techniques were supplanted by hidden Markov models which were a superior mathematical framework for incorporating statistical modeling techniques. However, DTW-based search has remained attractive and has been used by researchers incorporating neural network outputs for ASR [2, 3], and more recently for scenarios where there is little, if any, training data to model new words [4, 5, 6].

One attractive property of DTW is that it makes no assumptions about underlying linguistic units. Thus, it is amenable to situations where there is essentially no annotated data to train a conventional ASR engine. In our research, we are interested in developing speech processing methods that can operate in such unsupervised conditions. While we are ultimately interested in learning underlying phonological units of new languages automatically, we are initially focusing on learning new words and word sequences, and performing search queries on large corpora of unannotated data in an unknown language. In the latter case, we have an example speech query pattern, we wish to find the top K nearest-neighbor (KNN) matches in some corpus of speech utterances. DTW is a natural search mechanism for this application, although, depending on the size of the corpus, there can be a significant amount of computation involved in the alignment process.

For aligning two time-series consisting of M and N elements or frames, DTW conservatively takes $O(MN)$ time to compute a match. If one pattern sequence is much longer than the other, $M \ll N$, (e.g., searching for a word in a long recording), then any individual match will be $O(M^2)$, but we will need to initiate multiple DTW searches (in the extreme, starting a new search at every frame in the N frame sequence), which makes the overall computation more like $O(M^2N)$. For very large N, (e.g., $N > 10^7$) this can be a considerable burden.

To solve this computational problem, several lower-bound algorithms have been proposed for DTW search in large databases [7, 8, 9]. The basic idea behind lower-bounded DTW is similar in concept to the use of the future underestimate incorporated into A^* graph search. To start, N lower-bound DTW estimates are quickly computed between the query pattern, Q and every possible speech segment, S, in the corpus of utterances. These lower-bound estimates are sorted into a queue. Then, the lowest lower-bound estimate is incrementally popped off the queue and the actual DTW alignment score is computed for that particular match. This step is repeated until the lowest estimate remaining in the queue is greater than the K^{th}-best DTW score. The K-best search can then be terminated.

The prior research in lower-bound estimates for DTW search has focused on the Euclidean distance as the pairwise distance match between two vectors. In our recent work with DTW, however, we have been using a representation based on a sequence of posterior probability vectors – posteriorgrams, because of their superior generalization across speakers compared to spectral-based representations [10, 11]. Posteriorgrams have been explored by many researchers and are typically produced by phonetic recognizers [3, 12, 13, 14]. In our research they are generated by a set of Gaussian mixture models (GMMs) that are trained on a corpus in an unsupervised fashion.

In using the posteriorgram representation, it has been found that the inner product between posteriorgram vectors produces superior results on a variety of DTW-based tasks [11, 15]. Thus, if we are to leverage the lower-bound concept for reducing DTW computation, we need to derive a lower-bound estimate method for inner product distances. In this paper, we describe the lower-bound estimate method that we have developed for inner-product distances, and prove that it is admissible for KNN-based DTW search. We then perform keyword spotting experiments and compare the result to previously reported results, and show that we can eliminate 89% of the DTW calculations without affecting performance.

In the remainder of the paper, we first review basic concepts and notations of DTW and posteriorgrams in Section 2. The derivation and proof of the proposed lower-bound estimate is given in Section 3, and a description of the KNN keyword search using DTW with the lower-bound estimate is given in section 4. Experimental results and discussion are reported in Section 5. Finally, we conclude and suggest some future research.

978-1-4577-0539-7/11/$26.00 ©2011 IEEE

2. BACKGROUND

In this paper we derive a lower-bound estimate for DTW based on Gaussian posteriorgrams. However, we believe that the estimate is valid for all posteriorgrams, such as the phonetic posteriorgrams used in [12, 13].

2.1. Gaussian Posteriorgram

The Gaussian posteriorgram is a feature representation of speech frames generated from a GMM. In our work, a D-mixture, unsupervised GMM, G, is trained from a set of unlabeled speech frames, $\vec{x}_1, \ldots, \vec{x}_N$. A posterior probability, $p_i^j = P(g_j|\vec{x}_i)$, can then be calculated for any speech frame, $\vec{x}_i$, for each Gaussian component $g_j \in G$. A speech frame, $\vec{x}_i$, can then be represented by a D-dimensional posterior probability feature vector, $\vec{p}_i = \{p_i^1, \ldots, p_i^D\}$, where $\sum_j p_i^j = 1 \quad \forall i$.

2.2. DTW on Gaussian Posteriorgrams

Consider the two posteriorgram sequences for a speech query, $Q = \{\vec{q}_1, \ldots, \vec{q}_M\}$, and a speech segment, $S = \{\vec{s}_1, \ldots, \vec{s}_N\}$, where $\vec{q}_i$ and $\vec{s}_j$ are D-dimensional posterior probability vectors. The local distance between $\vec{q}_i$ and $\vec{s}_j$ can be defined by their inner product as $d(\vec{q}_i, \vec{s}_j) = -\log(\vec{q}_i \cdot \vec{s}_j)$. Given a particular point-to-point alignment warp, $\phi = (\phi_q, \phi_s)$, of length K_ϕ between Q and S, the associated alignment score, $A_\phi(Q, S)$, is based on the sum of local distances

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

where $1 \leq \phi_q(k) \leq M$ and $1 \leq \phi_s(k) \leq N$. The overall best alignment score, $\text{DTW}(Q, S) = \min_\phi A_\phi(Q, S)$.

If all possible warping paths, ϕ, are considered between Q and S, then there are $O(\text{MN})$ inner-product distances that will need to be computed. In order to eliminate unreasonable warping paths, a global path constraint is usually used to keep the warping paths between Q and S from being too far out of alignment [1]. This can be accomplished, for example, by ensuring that $|\phi_q(k) - \phi_s(k)| \leq r$ so that the warp will keep local distances within r frames of each other along the entire alignment.

3. A LOWER-BOUND ESTIMATE FOR DTW

Given two posteriorgram sequences, Q, and S, we can determine a lower-bound of their actual DTW score by first deriving an upper-bound envelope sequence, $U = \{\vec{u}_1, \cdots, \vec{u}_M\}$, where $\vec{u}_i = \{u_i^1, \cdots, u_i^D\}$ and $u_i^j = \max(q_{i-r}^j, \cdots, q_{i+r}^j)$. Note that the variable r is the same as it used for the DTW global path constraint, and that, in general, $\sum_{j=1}^D u_i^j \geq 1$. U can thus be viewed as a D-dimensional windowed maximum envelope derived from Q. Figure 1 illustrates an example of U on one dimension of a Gaussian posteriorgram. A lower-bound DTW score between two posteriorgrams, Q and S, can then be defined as

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

where $l = \min(M, N)$. Note that the computational complexity of computing $L(Q, S)$ is only $O(l)$.

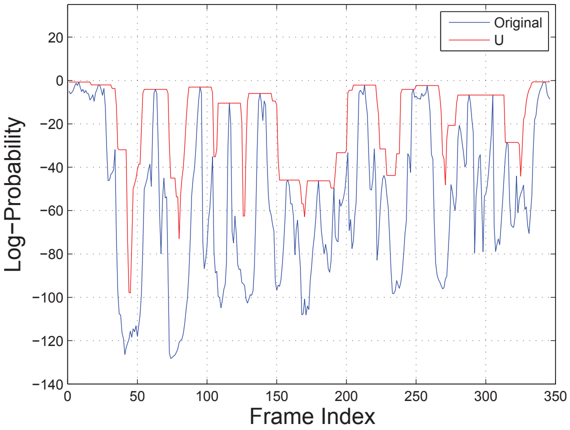

Fig. 1. Example of a 1-dimensional upper-bound envelope sequence (red) compared to the original posteriorgram (blue) for $r = 8$.

To prove that $L(Q, S) \leq \text{DTW}(Q, S)$ for posteriorgram sequences Q and S, we follow the strategies that are used in [7, 8]. By expanding both terms, we wish to show that

$$\sum_{i=1}^{l} d(\vec{u}_i, \vec{s}_i) \leq \sum_{k=1}^{K_\phi} d(\vec{q}_{\phi_q(k)}, \vec{s}_{\phi_s(k)})$$

where ϕ is the best warping path that produces $\text{DTW}(Q, C)$. The right hand side (RHS) can be further split into two parts

$$\sum_{i=1}^{l} d(\vec{u}_i, \vec{s}_i) \leq \sum_{k \in \text{MA}} d(\vec{q}_{\phi_q(k)}, \vec{s}_{\phi_s(k)}) + \sum_{k \in \text{UM}} d(\vec{q}_{\phi_q(k)}, \vec{s}_{\phi_s(k)})$$

where MA denotes a matched set containing exactly l warping pairs, while UM corresponds to an unmatched set that includes all remaining warping pairs. We construct the matched set as follows. For the i^{th} term on the left hand side (LHS), a warping pair $(\phi_q(k), \phi_s(k))$ from the RHS is selected into MA if $\phi_s(k) = i$. If there are multiple warping pairs from the RHS with $\phi_s(k) = i$, we select the pair with smallest $\phi_q(k)$ (although it only matters that one is selected). Note that there are always enough pairs to select into the matched set since $l \leq K_\phi$. By following this construction rule we ensure that the size of the matched set is exactly l so that each term on the LHS is matched exactly once by an element in the matched set. Based on the definition of the inner-product distance, all terms on the RHS are positive. Thus, all terms in UM can be eliminated if we can prove that the LHS is less than the terms in MA.

Consider an individual warping pair in MA, $(\phi_q(k), \phi_s(k))$, as it relates to the i^{th} term on the LHS, $d(\vec{u}_i, \vec{s}_i)$. By expanding the distance function back into the negative log inner-product, the inequality we need to prove becomes

$$\sum_{i=1}^{l} -\log(\vec{u}_i \cdot \vec{s}_i) \leq \sum_{i \in \text{MA}} -\log(\vec{q}_{\phi_q(i)} \cdot \vec{s}_{\phi_s(i)})$$

Since both sides now have the same number of terms, the inequality holds if each term on the LHS is less than or equal to the corresponding term on the RHS. By eliminating the log and examining only the individual dot product terms, we therefore need to show

$$\vec{u}_i \cdot \vec{s}_i \geq \vec{q}_{\phi_q(i)} \cdot \vec{s}_{\phi_s(i)}$$

```
Algorithm 1: KNN Search with Lower-Bound
Data: $Q$,$U$ and $C$
Result: $RL$ containing $k$ most possible utterances having
        keyword $Q$
begin
    for each utterance $c \in C$ do
        for each segment $s \in c$ do
            $lb = \text{ComputeLB}(U, s)$
            $PQ$.push($[lb, s]$)
    $KthBest = MaxFloat$
    while $PQ \neq \emptyset$ AND ($|RL| < k$ OR
    $PQ.top.lb < KthBest$) do
        $[lb, s] = PQ.top$
        $v = \text{DTW}(Q, s)$
        $c = \text{FindC}(s)$
        UpdateC($c, s, v$)
        if $c \in RL$ then UpdateRL($RL$)
        else $RL$.add($c$)
        if $|RL| \leq k$ then $KthBest = \text{FindMax(RL)}$
        else $KthBest = \text{FindKthMax(RL)}$
        $PQ$.pop()
```

Note that because of the way the matched set was selected, $\phi_s(i) = i$ so that $\vec{s}_i = \vec{s}_{\phi_s(i)}$. Since the DTW global path constraint r limits the warping pair so that $|\phi_q(i) - \phi_s(i)| \leq r$ we can also say $|\phi_q(i) - i| \leq r$, or $i - r \leq \phi_q(i) \leq i + r$. We can therefore see from the definition of u_i^j, that $u_i^j \geq q^j_{\phi_q(i)}$ so that our inequality holds: $L(Q, S) \leq \text{DTW}(Q, S)$.

One special property of the posteriorgram is that the summation of posterior probabilities from all dimensions should be one. Thus, the lower-bound could be trivial if $\vec{u}_i \cdot \vec{s}_i \geq 1$ because on the RHS $\vec{q}_{\phi_q(i)} \cdot \vec{s}_{\phi_s(i)} \leq 1$. However, if let $u_{max} = \max(u_i^1, \ldots, u_i^D)$, the LHS can be written as

$$\vec{u}_i \cdot \vec{s}_i \leq u_{max} \cdot \sum_{j=1}^{D} s_i^j = u_{max}$$

since $\sum_{j=1}^{D} s_i^j = 1$. Since we also know that $u_{max} \leq 1$ we can see that the lower-bound estimate is not the trivial case.

4. KNN SEARCH WITH LOWER-BOUND ESTIMATE

In order to determine the K nearest neighbor (KNN) segmental matches to a spoken keyword query, the default approach is to consider every possible match for all possible segments in each corpus utterance. By incorporating the lower-bound estimate described previously, we can find the top KNN matches much more efficiently, as shown in pseudo-code in Algorithm 1.

The basic idea of the algorithm is to use the lower-bound DTW estimate to prune out utterance segments whose lower-bound DTW estimates are greater than the K^{th} best DTW score. In Algorithm 1, the function *ComputeLB* calculates the lower-bound DTW estimate between the spoken keyword query, Q, and every possible utterance segment, S, using the upper envelope U. All utterance segments and their associated lower-bound estimates are stored in a priority queue ranked by the lower-bound estimate.

During KNN search, the algorithm begins from the top (smallest estimate) of the priority queue and calculates the actual DTW distance between the spoken keyword query and the associated segment. After using the function *FindC* to locate the utterance containing the current segment, the function *UpdateC* updates the best DTW distance in that utterance. Then, the function *UpdateRL* updates the result list if the best DTW score in the current utterance changes. If the result list does not contain the current utterance, the current utterance is added into the result list. Finally, if the size of the result list is less than or equal to K, the K^{th} best is set to the maximum value of the associated DTW score of the utterances in RL. If the size of the result list is greater than K, then the K^{th} best is set to the K^{th} maximum value in RL. The search algorithm ends if K possible utterances are in the result list and the lower-bound estimates of all remaining segments in the priority queue are greater than the K^{th} best value.

5. EVALUATION

Since we have shown that the lower-bound estimate results in a KNN-DTW search that is admissible, we wish to measure how much computation can be saved with the lower-bound estimate. In order to do this, we have chosen to duplicate previously reported keyword spotting experiments that we have performed on the TIMIT corpus using posteriorgram-based DTW search [10].

5.1. Keyword Spotting Task

The unsupervised keyword spotting experiment was performed on the TIMIT dataset, using the standard training set of 3,696 utterances and test set of 944 utterances. Since the lower-bound estimate does not require any parameters, no development set was used for tuning. A conventional MFCC-based spectral representation was used to generate 13 MFCC's every 10ms using a 25ms analysis frame. A 50 component GMM was created in an unsupervised fashion on the training set. The GMM was used to subsequently represent with 10ms frame with a 50-dimensional posteriorgram.

For the keyword spotting experiments, 10 keywords were randomly selected and one example for each keyword was extracted from the training set. The keyword spotting task was to find the K best matching utterances in the test set that contained the keyword. Matching was performed using segmental DTW on each utterance. More specifically, to compare the spoken keyword query with a test utterance, a sliding window with the size equal to the length of the keyword was applied to the test utterance to constrain the DTW search region. The sliding window gradually moved (one frame forward at a time) from the beginning frame of the test utterance to the end frame, and a series of DTW matches was performed to locate the best matching segment containing the keyword query [10]. The score for a test utterance containing the keyword query corresponds to the smallest DTW score obtained in that utterance.

5.2. Keyword Spotting Results

Since the lower-bound estimate is admissible, the results obtained by the new KNN method are the same as have been reported earlier where we achieved 14.6% equal error rate (EER) [10]. Of greater interest, however, is the amount of computation we can eliminate with the lower-bound estimate.

Figure 2 summarizes the amount of computational savings which can be achieved with the lower-bound estimate. The figure plots the average DTW ratio (scaled by log with base 10) against the size of K in the KNN search for several different DTW path constraint settings ($r = 1, 3, 5, 7, 9$). The average DTW ratio is

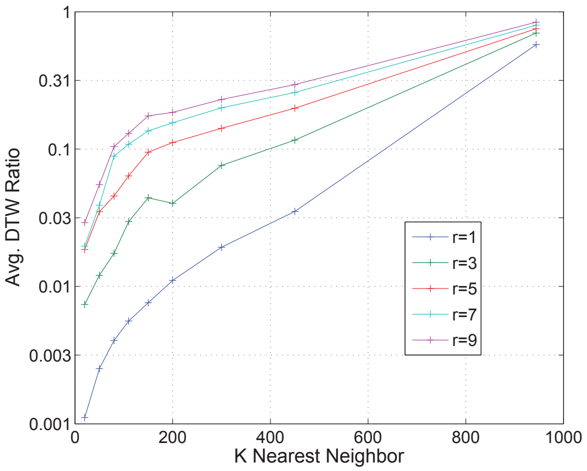

Fig. 2. Average DTW ratio against KNN size for different global path constraints, r.

the ratio of the number of utterance segments that require a DTW score to be computed divided by the total number of segments for a particular keyword search, averaged across all 10 keyword queries. For $r = 1$, for example, an exact DTW computation is needed for only 0.11% of all possible segments when K=10. In our prior experiments, we achieved a minimum EER for $r = 5$ [10]. With the current framework we achieve the same EER for K=200, and require only 11.2% of exact DTW scores to be computed compared to our previous results. Finally, it is interesting to note that even when K=944 (i.e., which finds all utterances in the test set), only 75% of the DTW calculations are needed since the lower-bound estimate prunes away undesirable segments in each utterance.

In terms of computation time, the results are quite dramatic. While our original DTW experiments required approximately 2 minutes to search for matches for the 10 keywords on a 200 CPU compute cluster, the new DTW-KNN method takes approximately 2 minutes on a single CPU or about 12s/query. Since the test set corpus corresponds to approximately 48 minutes of speech, this translates to approximately 14 seconds/query/corpus hour/CPU.

6. CONCLUSION AND FUTURE WORK

In this paper, we present a lower-bound estimate for DTW-based methods that uses an inner-product distance metric such as for a posteriorgram representation. Given a query posteriorgram and a test posteriorgram, the lower-bound is obtained by calculating the inner-product distance of the upper envelope of the query posteriorgram against the test posteriorgram. The lower-bound underestimates the actual DTW score between the query and test posteriorgrams, which provides an efficient pruning mechanism for KNN search. Based on the experimental results in a spoken keyword spotting task, the KNN-DTW search can eliminate 89% of DTW calculations while producing the same detection performance as the baseline system with no pruning.

Although we have already demonstrated the lower-bound estimate works effectively in a DTW-based keyword spotting application, it should be equally effective in all uses of DTW-based search. For example, we plan to quantify its performance in unsupervised pattern discovery applications that we are also exploring [6, 11]. This task essentially involves convolving a corpus of utterances against itself to find re-occurring patterns. In unsupervised contexts where there is no training data, a segmental DTW framework has been found to be effective, although a significant amount of computation is required. Recent research results have shown that a tremendous amount of computation can be avoided [14]. Based on the computational savings we have achieved here, we expect the lower-bound estimate will be very effective on this task, and has the added advantage of being admissible for KNN search.

Finally, since the lower-bound calculation can be easily parallelized, we plan to examine other computing architectures such as GPU computing to further speed up the entire algorithm.

7. REFERENCES

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

[2] C.-H. Lee, F. K. Soong, and K. K. Paliwal, *Automatic Speech and Speaker Recognition: Advanced Topics*, Springer, 1996.

[3] H. Hermansky, D. P.W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000.

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

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

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

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

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

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

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

[11] Y. Zhang and J. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *Proc. ICASSP*, 2010.

[12] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. ASRU*, 2009.

[13] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, "NLP on spoken documents without ASR," in *Proc. EMNLP*, 2010.

[14] A. Jansen, K. Church, and H. Hermansky, "Towards spoken term discovery at scale with zero resources," in *Proc. Interspeech*, 2010.

[15] A. Asaei, B. Picart, and H. Bourlard, "Analysis of phone posterior feature space exploiting class specific sparsity and MLP-based similarity measure," in *Proc. ICASSP*, 2010.