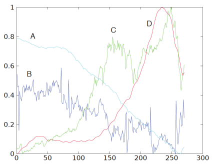
sDTW: COMPUTING DTW DISTANCES USING LOCALLY RELEVANT CONSTRAINTS BASED ON SALIENT FEATURE ALIGNMENTS

K. Selçuk Candan Arizona State University
Maria Luisa Sapino University of Torino
Xiaolan Wang Arizona State University
Rosaria Rossini** University of Torino

This work is supported by the NSF Grant 1043583 MiNC: NSDL Middleware for Network- and Context-aware Recommendations.

Time Series Search and Classification

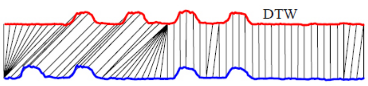
- Many applications generate and/or consume temporal data
- Querying and clustering of sequences and time series have been core data operations in many application domains
  - e.g. speech recognition, intrusion detection, finance



Sample economic index time series: A and B are similar to each other and different from the others (similarly for the pair C and D)

DTW: A (Common) Solution for Time Series Search

Dynamic time warping (DTW [1]) is a common technique for comparing sequences or time series by searching for optimal alignments.



The method is called "time warping" since it compresses or expands time in order to find the best mapping among points in two series.

[1]Richard W. Hamming. Error-detecting and error-correcting codes, Bell System Technical Journal 29(2): 147-160, 1950.

4

Warp Path

Let us be given two sequences or time series, $X = (x_1, x_2, \ldots, x_N)$ and $Y = (y_1, y_2, \ldots, y_M)$, where x_i and y_j are from the same domain D, and let $\Delta()$ be a distance function for comparing elements in D. An alignment from X to Y is described in terms of a *warp path* $W = (w_1, w_2, \ldots, w_K)$, where

- $max(N, M) \leq K \leq N + M$,
- $w_1 = (1, 1)$,
- $w_K = (N, M)$, and
- $w_l - w_{l-1} \in \{(1,0), (0,1), (1,1)\}$.

5

DTW distance

The *overall distance* of a given warp path, $W = (w_1, w_2, \ldots, w_K)$, between time series X and Y is defined as

$$\Delta(W) = \sum_{l=1}^{K} \Delta(x_{w_l[1]}, y_{w_l[2]}).$$

6

Optimal alignment

- An *optimal alignment* is defined as a warp path over the time series *X* and *Y* with the minimum overall distance over all possible warp paths.
- The goal of DTW algorithm is to find the *optimal* alignment between *X* and *Y*; in other words, the DTW
- distance between *X* and *Y* is defined as
- *ΔDTW(X, Y) = min{Δ(W) |Wis a warp path for X* and *Y}*.
- Note that the DTW distance is symmetric, but does not necessarily
- satisfy the triangular inequality – thus, it is not a metric.

DTW distance

The DTW distance between *X* and *Y* is defined as

- $\Delta DTW(X, Y) = min\{\Delta(W) \mid W$ *is a warp path for X* and *Y*$\}$.
- Note that the DTW distance is symmetric, but does not necessarily satisfy the triangular inequality – thus, it is not a metric.

Dynamic programming DTW computation

Testing all possible warp paths for *X* and *Y* would be prohibitively expensive

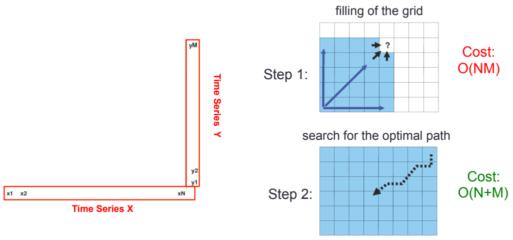
The DTW distance is also commonly computed by leveraging the underlying recursive nature of the distance function:
Let $X(1 : i)$ denote the *i*-length prefix of *X*,
$Y(1 : j)$ denote the *j*-length prefix of *Y*, and
$D(i, j)$ be defined as $\Delta DTW(X(1 : i), Y(1 : j))$.

$$D(i, j) = min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\}+\Delta(x_i, y_j).$$

Consequently, $\Delta DTW(X, Y) = D(N,M)$ and the corresponding optimal warping path *Wopt* can be identified using a dynamic programming algorithm that fills the $(N+1)$°—$(M+1)$ matrix,

DTW Grid and Warp Path

Given two time series, X and Y, of length N and M, alternative warping strategies can be compactly represented as warp paths on an NxM DTW grid.

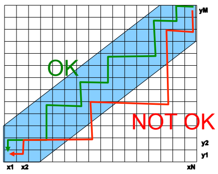


10

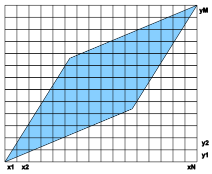
Reducing the Cost of DTW

To reduce the O(NxM) cost of filling the grid various heuristics impose constraints on the grid regions through which the warp paths can pass.

Sakoe-Chiba band [2]



Itakura parallelogram [3]

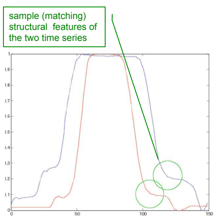


[2] Dynamic Programming Algorithm Optimisation for Spoken Word Recognition, 1978
[3] F. Itakura. Minimum prediction residual principle applied to speech recognition, 1975

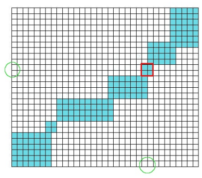
11

sDTW: Key Observation

Time series often carry temporal features that can be used for identifying locally relevant constraints to eliminate redundant work in an adaptive manner.



Two time series



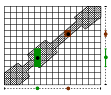
Adaptive constraints on the DTW grid

12

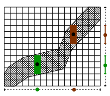
Proposed Adaptations

sDTW leverages salient alignment evidences to improve the effectiveness of the pruning constraints.

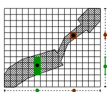
fixed core&adaptive width (fc,aw): the width of the band is adapted



adaptive core&fixed width (ac,fw): the core of the band is not necessarily on the diagonal



adaptive core&adaptive width (ac,aw): both the core and the width of the band are adapted



13

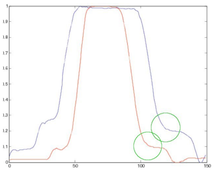
Overview of the sDTW Process

- **Step 1:** Search for salient temporal features of the input time series.
- **Step 2:** Find consistent alignments of a given pair of time series by matching the descriptors of the salient features.
- **Step 3:** Use these alignments to compute locally relevant constraints to prune the warp path search.

14

sDTW Step 1: Searching for Salient Temporal Features

Key question: How can we locate robust local features of a given time series?



15

Scale-Invariant Feature Transform

To search for robust temporal features, we adapt the 2D scale-invariant feature transform (SIFT) in a way that captures characteristics of 1D time series.

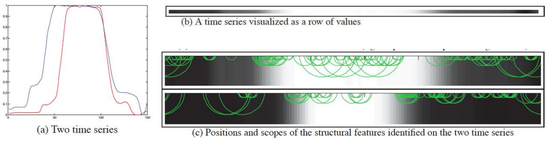
SIFT [*] is a computer vision algorithm used to detect and describe local features that are invariant to
- image scaling,
- translation,
- rotation, and
- different illuminations and noise.

in 2D images.

[*]D. G. Lowe. Object recognition from local scale-invariant features. In International Conference on Computer Vision,1999

(1D) Scale-Invariant Temporal Feature Transform

The proposed temporal feature extraction algorithm identifies
- positions and scopes of the salient temporal features and
- their feature descriptors



(a) Two time series

(b) A time series visualized as a row of values

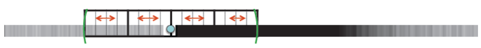
(c) Positions and scopes of the structural features identified on the two time series

(1D) Scale-Invariant Temporal Feature Transform

The proposed temporal feature extraction algorithm identifies
- positions and scopes of the salient temporal features and
- their feature descriptors

- Steps of the process:
  - Step 1.1: Scale-space extrema detection
    - Features are searched at multiple temporal scales
  - Step 1.2: Temporal feature descriptor creation
  - Step 1.3: Feature filtering and localization
    - Poorly differentiated features are eliminated

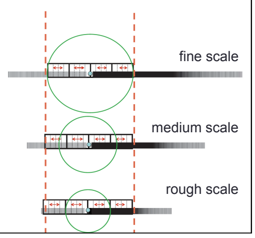
Salient Temporal Feature Descriptor



Once a salient feature is found, 2xr many gradients
- describing the magnitudes of changes at different distances from the center point of the temporal feature

are used for describing it.

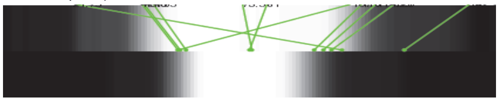
Note that the same descriptor size would cover different temporal ranges at different time scale.



sDTW Step 2: Searching for Consistent Alignments

This step involves

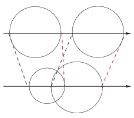
- Step 2.1: Temporal feature matching (Euclidean based comparison of descriptors)



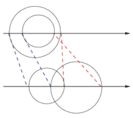
- Step 2.2: Temporal inconsistency removal

Temporal Inconsistencies

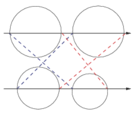
Temporal inconsistencies are observed when the start and end points of matching features do not conform to the same order in both time series



example 1



example 2



example 3

Ranking Feature Matches

A match is good if it implies good temporal alignment and similarity:

- Alignment ⟶ feature pairs with large temporal lengths and which are close to each other in time;

$$\mu_{align}(f_i, f_j) = \frac{(scope(f_i) + scope(f_j))/2}{1 + \mid center(f_i) - center(f_j) \mid}$$

- *Similarity* ⟶ pairs of features that have both similar descriptors and similar average amplitudes;

$$\mu_{sim}(f_i, f_j) = \frac{\mu_{descr}(f_i, f_j)}{\mu_{descr,min}} x(1 - \triangle_{amp}(f_i, f_j))$$

- *Combined score* ⟶ both good alignment and high similarity.

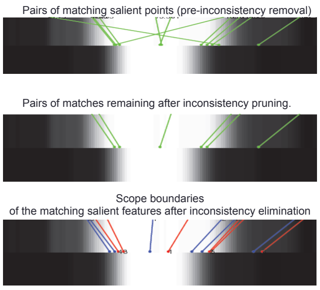
$$\mu comb(f_i, f_j) = 2x \frac{\mu_{align}(f_i, f_j) \times \mu_{sim}(f_i, f_j)}{\mu_{align}(f_i, f_j) + \mu_{sim}(f_i, f_j)}$$

22

Inconsistency Pruning

Given two time series,

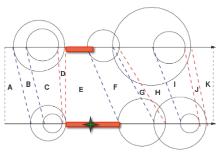
1. we consider the pairs of matching salient features in descending order of combined scores and
2. we prune those pairs whose boundaries imply inconsistent ordering against pairs that have been considered earlier.



23

sDTW Step 3: Searching for Locally Relevant DTW Constraints

Width Adaptation:



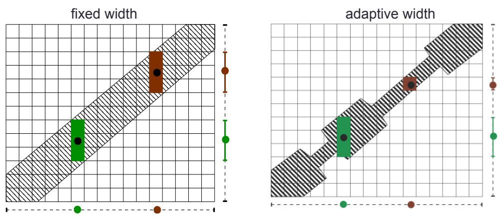
Consistently aligned features partition two time series into intervals

Thus, for each time point, we can adapt the width of the DTW band based on the lengths of the corresponding intervals

24

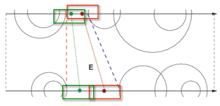
Adaptive Width Constraints

Adaptive width constraints use the widths of the resulting intervals to choose a different locally relevant width for each point.



sDTW Step 3: Searching for Locally Relevant DTW Constraints

Core Adaptation:

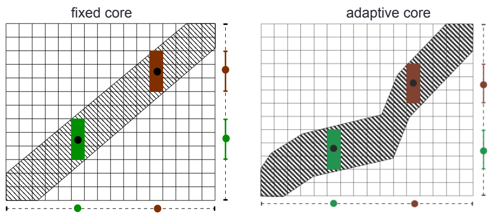


Each point on one time series has a roughly corresponding point on the other time series.

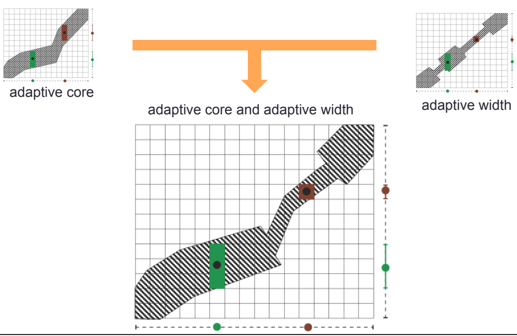
Thus, we can center the search band around these candidate points.

Adaptive Core Constraints

The core follows a path that reflects the candidate alignments implied by the salient features.



Adaptive Core & Adaptive Width Constraints



Complexity of DTW vs. sDTW

Time complexity for computing optimal DTW distance

- filling the N×M DTW grid: O(NM)
- identifying the optimal warp path: O(N+M)

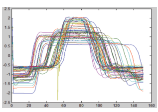
Time complexity for sDTW distance:

- extracting salient features : O(S(N+M)),
  - where S is the number of time scales considered
- finding matching salient feature pairs and pruning inconsistent pairs : O(UV),
  - where U<<N and V<<M are the number of features in the two time series
- Time for (partial) filling of the DTW matrix: O(ρNM),
  - where ρ<=1 is the selectivity of the locally relevant constraints

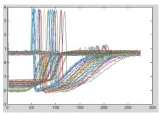
Evaluation Setup

We considered three diverse data sets:

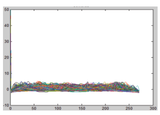
Gun Data Set



Trace Data Set



50Words Data Set



Data Set	Length	# of Series	# of Classes
Gun	150	50	2
Trace	275	100	4
50Words	270	450	50

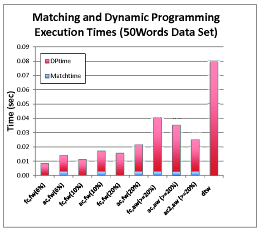
Experiment Setup

Hardware/Software:

- Intel Core 2
- Quad CPU 3GHz machine, 8Gb RAM
- Ubuntu 9.10(64bit)
- Matlab 7.8.0

For the baseline (FC&FW) schemes we used the Matlab code of Sakoe-Chiba [**].

Execution Time Analysis



The overhead of the matching step (for the adaptive algorithms) is negligible.

Effectiveness Criteria

In order to assess the effectiveness of various DTW algorithms, we use the following measures:

- Top-k Retrieval Accuracy

$$acc_{ret}(k) = avg\frac{|top_{dtw}(X,k) \cap top_{*}(X,k)|}{k}$$

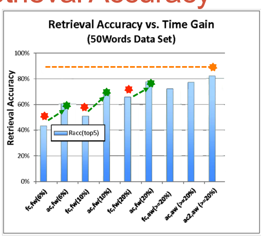
- Distance accuracy

$$err_{dist} = avg\frac{\triangle_{*}(X,Y) - \triangle_{DTW}(X,Y)}{\triangle_{DTW}(X,Y)}$$

- Top-k Classification accuracy

$$acc_{cls}(k) = avg\frac{|labels_{dtw}(X,k) \cap labels_{*}(X,k)|}{|labels_{dtw}(X,k) \cup labels_{*}(X,k)|}$$

Top-k Retrieval Accuracy



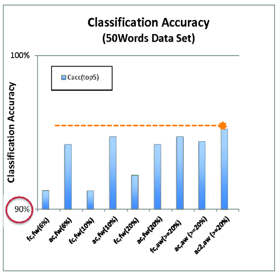
fixed core&fixed width (fc,fw) ⇒ larger widths imply more accurate results

adaptive core&fixed width (ac,fw) ⇒ provides significant gains in accuracy

adaptive core&adaptive width (ac,aw) ⇒ accuracy is boosted further

34

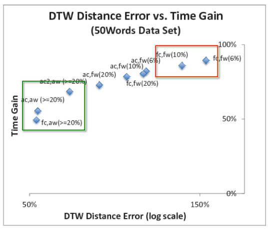
Classification Accuracy



Similarly to the top-k retrieval accuracy results, adaptive core and adaptive width algorithms provide the best classification accuracy.

35

Distance Accuracy



The figure confirms the accuracy results by focusing on the errors in the DTW estimates provided by the various algorithms

36

Conclusions

We recognize that the time series that are being compared often carry temporal evidences that can help prune unnecessary work during dynamic time warp (DTW) distance computation.

We have proposed three constraint types based on different assumptions on the structural variations in the data set.

Experiment results have shown that the proposed locally-relevant constraints based on salient features help reduce time in DTW distance estimations without negatively effecting accuracy.

Q&A

Thank you.