8-16-2013

Regularization of Dynamic Time Warping Barycenter Averaging, with Applications in Sign Classification

John A.W.B. Costanzo

Regularization of Dynamic Time Warping Barycenter Averaging, with Applications in Sign Classification

John A.W.B. Costanzo

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science *in* Applied & Computational Mathematics

School of Mathematical Sciences

College of Science

Rochester Institute of Technology

Rochester, NY

August 16, 2013

Committee Approval:

Dr. Nathan Cahill — Date
Thesis Advisor

Dr. Matthew Coppenbarger — Date
Associate Department Head

Dr. Raluca Felea — Date
Acting Program Director

Dr. Raja Kushalnagar — Date
Committee Member

Dr. David Ross — Date
Committee Member

Abstract

Sign language synthesis is a useful tool in addressing many of the issues faced by deaf communities. Sign languages are as different from spoken languages as spoken languages are from each other, and hence deaf persons raised learning sign language are not automatically proficient in communicating in written language. Existing methods of generating signing avatars are clunky and often unintuitive; hence the ability to classify gestures common to sign language using only data recorded by video would simplify the process dramatically.

Methods of gesture classification require a way to compare time series, and often (in particular, for k-means clustering [6]) require a notion of "average" or "mean". However, computing the average of a collection of time series is difficult. Time series infer no meaning from the index of a particular frame; only the order, and not the time index, of features confer meaning. Dynamic time warping [12] was developed as a similarity measure between time series, but does not in itself provide a method of averaging. Recently, a method of averaging called DTW Barycenter Averaging (DBA) was developed [11] that is consistent with dynamic time warping. This method produces results suitable for classification and clustering of time series data, and is based on minimizing the within group sum of squares (WGSS) of the data.

Because dynamic time warping is time scale invariant, the average is not unique; other warpings of an average may also be averages. We propose a modification to DBA that allows for more flexibility in choosing the time scale of the resulting average. Time penalized DBA (TBA) adds a cooling regularization term to WGSS, making the problem well-posed. The regularization term penalizes the amount of total warping between the average and each other time series; hence features in the average appear closer to the average *time* at which they appear in the collection. We cool the regularization term to prevent it from altering the solution in undesirable ways.

Time penalized DBA is an effective method to average a collection both spatially and temporally, and also reduces the algorithm's sensitivity to initial guess. Unfortunately, the extra parameters it requires make its use more complicated. We will show for a selection of parameters that TBA performs favorably over classical DBA on both artificial signals and on data captured from videos of signs from American Sign Language.

Acknowledgments

This thesis is dedicated to my father William, for his philosophy and guidance over the years. His outlook on life shaped who I am today, and without him this day would not be possible.

I also owe this day to my family, and in particular my Aunt Dolores and her husband Robert, for their efforts and support through my father's illness and passing.

I would like to thank my mentor, Dr. Nathan Cahill, and his colleagues, Dr. Sonja Lopez Alarcon, Dr. Raja Kushalnagar and Dr. Harvey Rhody for allowing me to work on this project. In particular I would like to thank my colleagues, Michelle Chung, Lee Wingfield, and Raghu Puppala for writing the script for and recording the sign language videos used in this thesis.

I also thank Dr. Cahill for his advice through this project and guiding me toward my final results.

I would also like to thank Dr. David Ross for giving me my first research project many years ago, for imparting to me his work ethic, and for participating in my thesis committee.

I also acknowledge Nicola Talbot of the University of East Anglia for the helpful tips in her slideshow on writing theses in LaTeX.

I also thank my girlfriend Xiao for her delicious cooking and moral support during the course of this project.

Contents

List of Figures

List of Tables

Introduction

Gesture recognition is an important topic in the fields of computer vision and machine learning because of its applications in childhood deaf education and deaf online anonymity. Sign language synthesis, whereby a sign is performed by a computer generated avatar, is a popular tool used in video games that teach deaf children sign language [17] and transcribing written language into signed language [5] for deaf persons who are not fluent in a written language. A signing avatar can also anonymously represent a deaf person who wishes to participate in online discussion forums but cannot communicate in a written language. To generate this avatar, it must be told how to produce the sign; current methods of inputting this data require clunky apparatuses or manually typed scripts. It is our hope that computer vision techniques can simplify this process by requiring only a video of the sign to be synthesized.

A variety of classical machine learning algorithms are applicable in gesture recognition, given a notion of "distance" between gestures, which are represented as time series. A distance between time series is more difficult to define than purely geometric data, because only the order of the elements in the time series, and not their exact time index, confers meaning. Dynamic time warping [12] (DTW) is a standard algorithm for computing the distance between time series and has been explored in, among other applications, signature recognition [12, 13].

Many distance-based learning algorithms can employ DTW, but many learning algorithms such as k-means clustering [6] require a method of averaging [11]. Computing the average of a collection of time series in a way that is consistent with DTW is nontrivial, because as the average is unknown, so too is the temporal alignment of each time series to the average. A recent method called DTW Barycenter Averaging [11] (DBA) was shown to give good solutions to the average time series problem. However, DBA is an iterative algorithm, and its solution is sensitive to the initial guess.

Although convergence results exist for DBA [11], it purports to solve an ill-posed problem as any time warping of an average may also be an average. We propose a modification of DBA that is less sensitive to the initial guess, and whose solution has a more uniform scaling with respect to the time scalings of the time series being averaged. That is, features present in the average will also appear at the average time in which they happen.

This thesis will address the issues present with averaging time series and propose an improvement to a method of finding the average of a collection of time series. This thesis is laid out as follows:

Chapter 1 provides an overview of the significance and methods of gesture recognition as they apply to deaf education and internet accessibility. We will also demonstrate the need for a robust method of averaging time series data in the use of clustering algorithms that require a method of averaging.

Chapter 2 provides an in-depth review of the literature behind existing methods of time series comparison. We present a review of dynamic time warping (DTW), a classic algorithm that finds the optimal registration of time series along with the associated distance between them. We also mention derivative dynamic time warping, which overcomes some of the anomalies that can result from DTW by comparing slopes and peaks in time series, instead of individual frames themselves.

Chapter 3 will propose our modifications to DTW and DBA. Our techniques are original because they provide a more well-posed solution than existing methods. Our new techniques are compared along with classical DBA to show that they produce more consistent results contrived data. We will discuss the effects of the parameters to our method by experimenting on artificial signals. We will also demonstrate some of the benefits that our modifications impart to the resulting average, including reduced sensitivity to the initial guess, and a more uniformly time scaled solution. The section, "Improvements to DBA", is the main proposal of this thesis.

Chapter 4 will present a performance comparison between our proposed algorithm and the preexisting DBA algorithm, as applied to sign language data captured by the Microsoft Kinect. We will also enumerate some of the more common gestures in American Sign Language (ASL) that are included in the data. We will show that our algorithm performs comparably to DBA at minimizing the within-group sum of squares of the data, and conclude that our algorithm is a suitable augmentation.

Chapter 1

Project Overview

1.1 Significance

Because of the differences between signed language and spoken language [7], it is a common misconception that deaf Americans who are brought up learning American Sign Language are automatically fluent in English. Books and programs designed to teach hearing children English assume some existing background in English as hearing children are exposed to spoken English from infancy. These same techniques do not apply as well to a child who has had no exposure to the language, and moreover cannot supplement their English education through everyday verbal communication.

As a result, deaf Americans who cannot easily communicate in writing have difficulty participating anonymously in online communities and reading online content. Signers who are not fluent in English who wish to participate in an online forum must either upload a video of themselves signing their response, which violates their anonymity, or find an interpreter to dictate their thoughts into text.

1.2 Sign Language Synthesis

Computer generated signing avatars have been investigated by several sources [4,5,17] to address the issues of childhood deaf education and online deaf anonymity. So far, these avatars have been synthesized either from written language description of the sign or

from data captured from bulky and expensive motion capture suits [10]. These systems would be improved if it were possible to mimic signs by estimating these linguistic features from recorded video.

Figure 1.1: Animation of the SignSmith avatar [5]

Figure 1.2: The CopyCat system is used for verification of sign reproduction, using colored gloves and accelerometers [17]

Signs in ASL can be described by their manual and nonmanual components [14, 15]. Manual features include the shape, location, movement, and orientation of the hand. As ASL is a very expressive language, the interaction of nonmanual features is more complex; facial expression is heavily emphasized in particular, but the movement of other body parts such as the elbows and shoulders may impart meaning as well.

It is the hand movement, or **gesture**, that is our primary focus. To simplify the image processing, we are capturing videos of signs using the Microsoft Kinect, seen in Fig. (1.3). The Kinect, in addition to its RGB camera, has an infrared depth camera which makes thresholding of actors easier. Moreover, the Microsoft Kinect Software Development Kit (SDK) [1], along with its open-source counterpart, OpenNI [2], automatically provide access to the location of joints such as the neck, palms, shoulders, and elbows, as illustrated in Fig. (1.4). The Microsoft SDK uses 20 joints, but OpenNI omits the ankles, wrists, and the center of the spine.

Figure 1.3: The Microsoft Kinect. [1]

Figure 1.4: A video frame captured by the Kinect using OpenNI.

1.3 Gesture Classification

Machine learning is the process of recognizing structures in data sets. An application of this is to determine what events are taking place based on measurements of observations of these events. Machine learning is primarily separated into two disciplines: *supervised* and *unsupervised* learning.

A supervised learning machine is given a set of data points, or *observations* that have already been given a label, or *classification*. When given a new data point, the machine will attempt to assign to it one of these classifications depending on which existing points it is closest to. A supervised learning machine can only provide a classification that another observation already belongs to.

Conversely, an unsupervised learning machine is given only a set of data points, but no information about their classification. The unsupervised learning machine must separate these points into *clusters* thereby imposing a structure on the unlabeled data. There also exist semi-supervised learning machines, which are given *some* information about the structure of the data but in which most of the data remains unclassified.

In either case, a method is required to compare the data points. This may be the Euclidean or hamming distance, but if the data points are time series of different lengths and time scaling, these traditional metrics are insufficient—in fact, they are not defined between sequences of different lengths.

For instance, two videos of signers tracing a circle with their hand may differ in which frame the signer has his or her hand at the top of the circle, and the number of frames it takes the signer to move from one particular point on the circle to another may also

differ. This is due to different signers signing at different speeds—some may even speed up and slow down differently. Further, the videos may not even have the same number of frames.

Dynamic time warping (DTW) has been around at least since 1971 [12] and is in widespread use in signature recognition. A warping path is a list of associations between time series; coordinates that are associated are interpreted to be the same feature of the time series, regardless of the time at which they happen. DTW is an algorithm that simultaneously computes both the optimum warping path between two time series and the total cost associated with this path. The total cost of the warping path is a semi-pseudometric on the space of all time series over the associated space; often the terms "dynamic time warping" and "DTW" are used to refer specifically to this cost.

Dynamic time warping is an effective metric when comparing time series exhibiting different speeds. However, it is complicated to define a notion of the *average* (a requirement for some clustering schemes such as k-means clustering) of a collection of time series in a space where no canonical time scale is established. Consequently, a common technique for gesture recognition is to model gestures as a Hidden Markov Model [16] instead. However, HMMs are computationally expensive to train and may be more powerful than is necessary for gesture recognition.

To provide more options for the choice of algorithm, a few methods for averaging a collection of time series have been proposed. Until recently, these have involved pairwise averaging using various ordering schemes. However, these methods are sensitive to the order in which the time series appear in the collection. DTW Barycenter Averaging [11] (DBA) solves this problem with an iterative technique, which depending on the initial guess produces superior results to preexisting pairwise methods. However, DBA is sensitive to the initial guess and may produce inconsistent results, as neglecting the time axis altogether makes the problem of averaging ill-posed. As a result, even if DBA does provide a good average, its solution is only unique modulo time shifts in the solution.

1.4 Summary

This thesis is part of a larger goal of improving accessibility and educational technology for the deaf. Signing avatars have applications in childhood deaf education, transcription of written language, and providing online deaf anonymity. Current signing avatars require clunky apparatuses or complicated manually typed input to reproduce a sign.

Computer vision techniques can simplify the process by requiring only a video of the sign for the avatar to mimic. This requires automated classification of the manual and nonmanual features of the sign. The hand movement, or gesture, is one such feature. While a variety of existing classification techniques can already be employed to gesture recognition, many other classification techniques require a way to compute the average of subcollections of data.

Improvements have been made in providing this average. While older methods were sensitive to the order of the collection, DTW Barycenter Averaging produces superior results to these methods and is order independent. However, DBA does leave room for improvement as it was developed with the intention of solving a still ill-posed problem. As a result, DBA is sensitive to initial conditions and is likely to find a local optimum at the expense of finding a global optimum.

Chapter 2

Existing Methods for Time Series Analysis

Time series analysis requires more complicated methods than that of simple data. It is not trivial to define a "natural" notion of distance between time series, as only the order of events is important and not the exact time at which an event happens. We will review dynamic time warping [12], a classic algorithm that provides a registration between time series, and uses this registration to compute a time independent "distance" between them. We will also review DTW barycenter averaging, [11] a method of averaging a collection of time series in a manner consistent with DTW. We will also discuss what is meant by "consistent with DTW".

2.1 Dynamic Time Warping

2.1.1 Motivation

Dynamic Time Warping (DTW) is a similarity measure that attempts to compare the overall features between time series while ignoring variabilities in speed between them. [11] To motivate the need for dynamic time warping, let us explore the drawbacks of a simpler and more well-known metric.

Let $A = \begin{bmatrix} a_1 & a_2 & \dots & a_{T_A} \end{bmatrix}$ and $B = \begin{bmatrix} b_1 & b_2 & \dots & b_{T_B} \end{bmatrix}$ be sequences with each a_i and b_i from a pseudometric space (E, δ). We will refer to δ as the **cost function** (of E),

E^T as the set of all sequences of elements of E of length T, and denote E^ω as the space of all such sequences over E.

The Euclidean distance between A and B (assuming $T_A = T_B = n$) is given by

$$d(A, B) = \sqrt{\sum_{k=1}^{n} \delta(a_k, b_k)^2}.$$

This is the simplest distance between two sequences, but it is not defined if the sequences are time series of different lengths. Even when comparing time series of the same length, the Euclidean distance is not desirable as it does not allow for events to happen at slightly different times.

For instance, the Euclidean distance would separate the vectors $\begin{bmatrix}1 & 1 & 100 & 1\end{bmatrix}$ and $\begin{bmatrix}1 & 100 & 100 & 1\end{bmatrix}$, and $\begin{bmatrix}1 & 1 & 100 & 1 & 1\end{bmatrix}$ is not comparable to either one. This is less of an issue for digital signals, but a better metric is needed to compare sequences of measurements of analog phenomena, called **time series**, which will commonly exhibit such variability as above.

In the scope of this project, such variability may result from two videos of the same sign, but performed at different speeds, or perhaps even with different accelerations. For instance, one signer may perform a sign by starting out slowly and accelerating toward the end; whereas another signer may start fast but slow down.

2.1.2 The DTW Algorithm

The DTW algorithm [9, 12, 13] finds a **warping path** between the time series being compared. A warping path is a sequence $\begin{bmatrix}w_1 & w_2 & \dots & w_K\end{bmatrix}$ such that the following properties hold:

1. $w_1 = (1, 1)$ and $w_K = (T_A, T_B)$ (*boundary conditions*)
2. If $w_k = (i, j)$ and $w_{k+1} = (i', j')$ then $|i' - i| \leq 1$ and $|j' - j| \leq 1$ (*continuity*)
3. $i' - i \geq 0$ and $j' - j \geq 0$; at least one of these inequalities is strict. (*monotonicity*)

We will refer to the elements w_k of the warping path as **associations** between coordinates.

The cost of the optimal alignment can be found by the recursive formula: [12]

$$D(A_1, B_1) = \delta(a_1, b_1) \tag{2.1}$$

$$D(A_1, B_j) = \delta(a_1, b_j) + D(A_1, B_{j-1}) \tag{2.2}$$

$$D(A_i, B_1) = \delta(a_i, b_1) + D(A_{i-1}, B_1) \tag{2.3}$$

$$D(A_i, B_j) = \delta(a_i, b_j) + \min \left\{ \begin{array}{l} D(A_{i-1}, B_{j-1}) \\ D(A_i, B_{j-1}) \\ D(A_{i-1}, B_j) \end{array} \right\}. \tag{2.4}$$

where $A_i = \begin{bmatrix} a_1 & \dots & a_i \end{bmatrix}$. The overall cost $DTW(A, B; \delta) = D(A_{T_A}, B_{T_B})$, and can be found using dynamic programming [13] using a matrix M; $M_{ij} = D(A_i, B_j)$ by updating from the top-left corner outward.

2.2 Derivative Dynamic Time Warping

Because classical DTW considers only the differences in height between the associations made, it may fail to find the "best" warping path in terms of aligning features. For instance, if a_i has a similar value to b_j, but a_i is part of a rising trend and b_j is part of a falling trend, DTW would consider the pairing (i, j) to be ideal, even though we would prefer not to associate a rising trend with a falling trend [8].

This gives rise to a phenomenon known as **singularities**; a point in one time series that is associated to many points in the other time series. Considering associations as edges in a graph, a singularity is a star graph $K_{1,p}$ where p is "large".

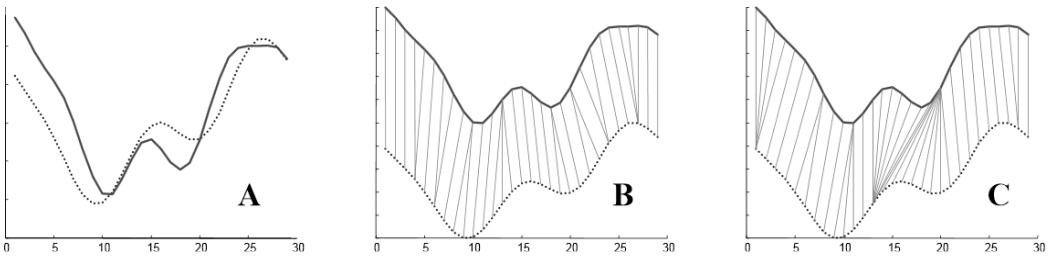


Figure 2.1: The features in these two signals have different y separations. We'd prefer the alignment in **B**, but DTW produces the alignment in **C**. [8]

Keogh & Pazzani [8] offer a practical solution to the problem of singularities. Derivative

Dynamic Time Warping (DDTW) uses the cost function $\delta(i,j) = (a'_i - b'_j)^2$, where x'_k denotes the estimated local time derivative of x at index k. Note now that δ is no longer a pseudometric on E, since it depends on the whole time series. (It is, however, a pseudometric on E^T.)

Keogh & Pazzani [8] recommend the following estimate for the local derivative:

$$x'_k = \frac{x_k - x_{k-1} + (x_{k+1} - x_{k-1})/2}{2} \tag{2.5}$$

which is simply the average of the slope of the line connecting x_k to its left neighbor, and the slope of the line connecting its left neighbor to its right neighbor.

2.3 The average time series problem

Regardless of the algorithm used to associate terms between time series, these associations can be used to compute an average for the purposes of clustering.

It is necessary to define what we mean by "average" however, in a space in which addition is not clearly defined. In particular, the regular arithmetic mean is not acceptable even between time series of the same length. For instance Fig. (2.2) shows different "averages" of three Gaussian pulses warped in time. The top left plot is the arithmetic mean; notice that it has three peaks in it, which is not at all representative of any of the three one-peaked time series being averaged if we consider these time series as the location of someone's hand as a function of time. Each of the time series being averaged represents a hand moving up, and then down; but the arithmetic average represents a hand repeating moving up and down twice!

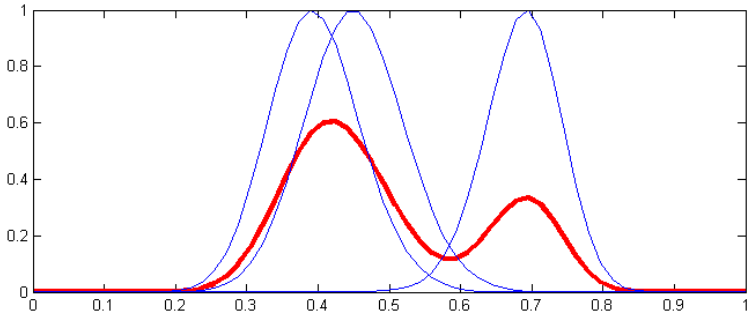


Figure 2.2: Arithmetic average of three Gaussian pulses that have been warped in time.

For our purposes, given a collection of time series $\mathbb{S} = \{S_1, S_2, \ldots, S_n\}$, the time series $C = \{c_1, c_2, \ldots, c_t\}$ is considered an average [11] of $\mathbb{S}$ if it minimizes

$$WGSS(X) = \sum_{k=1}^{n} d(X, S_k)^2 \tag{2.6}$$

among all time series $X \in E^\omega$. WGSS is an abbreviation for **Within Group Sum of Squares** and is often called simply **inertia**. Particularly since information about the length of C is unavailable, finding an exact solution to Eq. (2.6) is intractable.

2.3.1 Pairwise averaging

Given two time series A and B and a warping path between them, there are two common ways to directly find an approximation to their average [11].

One coordinate by association assigns to the average the center of each association. That is, if $w_k = (i, j)$ then $c_k = (a_i + b_j)/2$. The main problem with this scheme is that the average time series will have at least the same length as the larger of A or B, and can potentially be almost double that [11]. When applied pairwise to a collection of time series, the consequence of this growth in length depends on the ordering scheme but could be exponential in the case of a tournament scheme.

One coordinate by connected component treats the warping path as a collection of edges in a bipartite graph between the coordinates of A and B. For each connected component in the graph, a single frame is added to the average time series equal to the barycenter of the vertices of that component. Contrary to the above, the length of the average in this case can only decrease, having as few as 2 coordinates in the extreme case [11], and being at most of length $\min\{T_A, T_B\}$.

Pairwise averaging using tournament or other ordering schemes are simple and in standard use, but their sensitivity to the ordering of the collection of time series makes them unattractive.

2.3.2 Average of a collection using DBA

Petitjean et al. [11] also proposes an iterative technique known as **DTW Barycenter Averaging** (DBA) for computing the average of a collection of time series in a manner that does not depend on the order in which they appear in the collection. Starting with an initial guess C, we compare C to each time series in $\mathbb{S}$ using DTW. For each coordinate c_i in C, we let c_i' be the barycenter of all of the frames in all of the time series in $\mathbb{S}$ that are associated with c_i. We refer the reader to [11] for a more comprehensive detailing of the algorithm.

There are many advantages to this scheme. First, the length of C remains constant at each iteration; its length is easily determined by choosing an initial guess that has the desired length. Further, while successive pairwise averaging fails to be associative, DBA is completely independent of the ordering of the time series. Further, DBA was shown to minimize Eq. (2.6) more effectively than standard pairwise schemes [11].

2.4 Summary

We have reviewed some existing techniques for analyzing collections of time series data. Simple metrics are insufficient because they are too time dependent. Dynamic time warping is a useful algorithm that provides a meaningful notion of distance between time series. However, because of the features it considers, DTW may perform suboptimally if time series data differ locally in their frames. Derivative DTW effectively deals with this problem by associating the slopes and peaks of time series instead of just their y-coordinates.

We have also reviewed averaging schemes that are consistent with DTW. Older pairwise schemes are outdated in time series cluster analysis because of their suboptimal results and sensitivity to the ordering of the collection. DTW Barycenter Averaging produces superior averages and achieves these results in a manner which does not depend on the order in which time series appear in the collection.

Chapter 3

Modifications to DTW and DBA

Dynamic time warping and DTW barycenter averaging do allow room for improvement. In this chapter, we will present a continuous formulation of the time series registration problem that does not restrict points in one time series from being associated with exactly one point in the other. We will also explore a technique for parallelizing the DTW algorithm. No speedup was observed by implementing this parallelization because of the overhead present in MATLAB's built-in parallelization features, but may be worth exploring in Open MP. However, we will demonstrate conversely that *DBA* does benefit from parallelization, and that doing so is fairly straightforward.

We will then provide a new cost function and a new family of cost functions that dynamic time warping can use in lieu of the standard sum-of-squared-differences. In particular, the time penalty method is a family of cost functions that adds a regularization term to an existing cost function. We will conclude by exploring the use of time penalty in the DBA algorithm and shows that it performs far better than DBA on synthetic signals in terms of sensitivity to initial guess, and in the time scaling of the resulting average.

3.1 A Continuous Formulation

Dynamic time warping treats time series as discrete sequences of events. In reality, while computation deals only with discrete data, these are often measurements of analog phenomena. While a correct warping in the continuous sense may associate a_i to a

point somewhere in between b_j and b_{j+1}, DTW is forced to choose one of them, with potentially suboptimal results.

Another way to register two time series is to think of them as discretizations of twice continuously differentiable functions $f, g : [0, 1] \to E$, where $f(t)$ is a canonical representation of an action, and $g(t)$ is a measurement of this action, where $f(t) = g(t-u(t))+\varepsilon(t)$ for some $u(t)$ and $\varepsilon(t)$. We assume that the differences between $f(t)$ and $g(t)$ are due solely to differences in time scale and not errors in measurement. Aligning these trajectories hence becomes the problem of minimizing $\|\varepsilon(t)\|$, or equivalently minimizing the functional

$$J(u, u') = \int_0^1 \left[\delta\left(f\left(t\right), g\left(t - u\left(t\right)\right)\right)\right]^2 dt + \left[J_{reg}(u)\right]^2 \tag{3.1}$$

where J_{reg} is a regularization term that guarantees the problem is well-posed. When $\delta(x, y) = |x - y|$, $E = \mathbb{R}$ and

$$J_{reg}(u) = \int_0^1 \alpha_1[u(t)]^2 + \alpha_2[u'(t)]^2 dt,$$

the Euler-Lagrange equations become

$$\begin{aligned} -\alpha_2 u''(t) + \alpha_1 u(t) = &- \left[f(t) - g(t - u(t))\right] g'(t - u(t)); \\ u(0) = 0; &u(1) = 0. \end{aligned} \tag{3.2}$$

This can be solved numerically, but a suitable interpolation technique must be applied to the data for f and g.

3.2 Parallelizing the DTW Algorithm

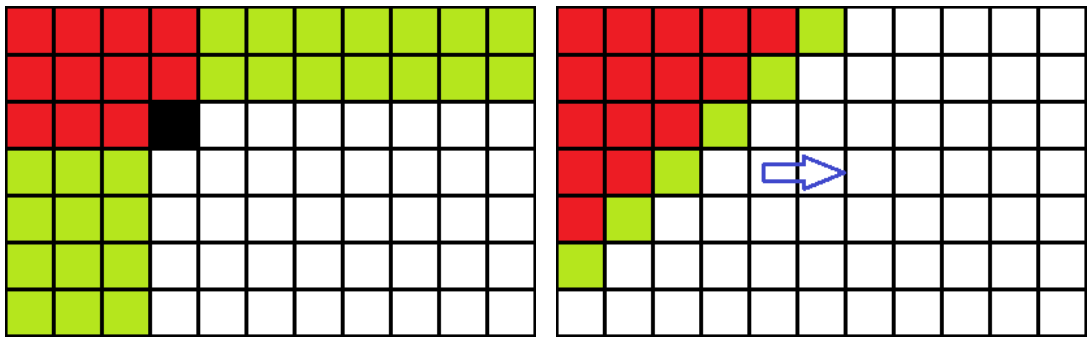
Recall from section 2.1.2 that the dynamic algorithm used to compute the costs of each alignment involves a matrix M, where $M_{ij} = D(A_i, B_j)$. Since the recursive formula at each entry depends on entries before it, it is not completely order independent. Hence, care must be taken in how parallelization is implemented.

Parallelization can be implemented along the *anti*-diagonals, beginning from the top left and working outward. Since each entry depends only on entries with smaller i and

Table 3.1: DBA computation time versus number of workers; 32 time series of length 60

Number of workers:	1	2	3	4	5	6	7	8
Computation time (s):	99.2	75.5	63.4	47.7	45.9	44.0	43.5	44.9

j values, the entries on a particular anti-diagonal are mutually independent and can be computed in parallel. Given enough workers, this means the task could be completed in $T_A + T_B$ sequential calculations (the number of anti-diagonals), instead of $T_A T_B$ (the number of entries).



(a) Red: Entries that black is dependent on; Green: Entries not dependent on black.

(b) Green: A maximal set of entries that can be computed in parallel; Red: The entries that must be computed first.

Figure 3.1: Filling the cost matrix in parallel

The cost of filling each entry of the matrix, in practice, does not warrant the extra overhead of MATLAB's built-in parallelization features; the time series we are working with are too small. In fact, parallelizing in this fashion was found to increase the time per iteration by over 200%.

However, significant speedup was observed when the averaging algorithm was parallelized. Each iteration requires associating the current average estimate to each of the time series being averaged; all of these associations can happen at once. Hence, in Algorithm 3.1, the **for all** loop on line 5 is done in parallel. The algorithm was timed using 1 through 8 workers on a sample dataset of 32 time series each containing 60 frames and is summarized in Table 3.1. The computations were performed on a laptop with a Core i7-740QM processor (6M cache, 1.73 GHz) and 4GB of RAM. The reason for the lack of speedup in increasing the number of workers above four is that the processor only has four physical cores.

3.3 Employing other cost functions in DTW

A variety of cost functions are possible when computing the cost matrix. One such cost function was introduced by [8] and discussed in Section 2.2 and uses as its cost function the Euclidean distance between the estimated time derivatives of the time series.

Another estimate of the time derivative is the centered difference formula,

$$x'_k = \frac{x_{k+1} - x_{k-1}}{2}. \tag{3.3}$$

The estimate at the endpoints is not defined by this formula, and is instead replaced by the estimate at the second and second-to-last endpoints. Eq. (3.3) is more likely to fail in the case of outliers because it uses fewer data points, but Eq. (2.5) produces unsatisfactory results when used in conjunction with DBA.

We also introduce a way of modifying existing cost functions by applying the transformation:

$$\hat{\delta}_\lambda(i,j) = \sqrt{[\delta(a_i, b_j)]^2 + \lambda^2 \left(\frac{i}{T_a} - \frac{j}{T_b}\right)^2}, \tag{3.4}$$

where λ is a weighting parameter. We call this the "time penalty" method as it adds a penalty to excessive time shifting. Similar to slope weighting [9, 13], a larger λ results in the algorithm tending closer to the diagonal. In MATLAB, we implement this by simply concatenating $linspace(0, \lambda, T_X)$ to the bottom row of both time series matrices (replacing T_X appropriately) prior to passing them into DTW, and using the standard Euclidean distance to compute the cost matrix.

When using Eq. (3.4) in dynamic time warping, we refer to it as **Time Penalized DTW** (TPD); or as a function we write $DTW(A, B; \hat{\delta}_\lambda)$.

3.4 Improvements to DBA

The minimizer of WGSS (Eq. (2.6)) is not necessarily unique, as any time warping of a mean may also be a mean. Often in contexts where DTW is being used, this is not an issue, but if it is desired that the features in the average also happen at the average time, modifications must be made to bring some time dependence back into the algorithm.

The choice of initial guess is important. Experimentation on contrived data showed that choosing an arbitrary $S_k \in \mathbb{S}$ provides convergence to a suitable solution. However, provided the time series in $\mathbb{S}$ are fairly close to one another under DTW, the solution given by DBA using naive DTW is very close in the Euclidean sense to the initial guess. In Figure 3.2, the time series chosen as the initial guess is the furthest to the right of all the time series in $\mathbb{S}$. Consequently, the peaks in the "average" actually happen *after* the peaks in all of the time series in $\mathbb{S}$.

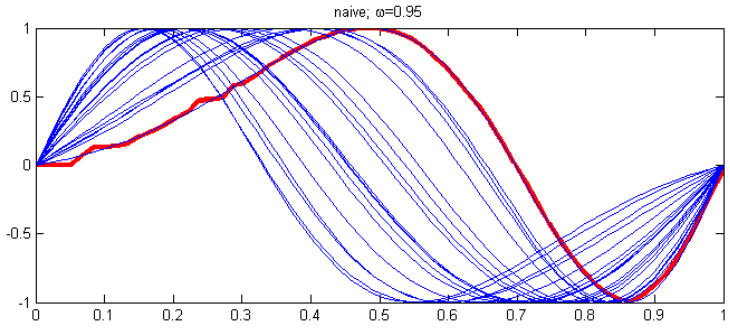


Figure 3.2: The average (red) computed using naive DTW

This is, in a sense, DTW doing its job properly; *when* a particular feature happens does not matter. However, it make more intuitive sense if the features in the average also happened at the average time at which all of the time series displayed this feature.

Conversely, the *one coordinate by association* scheme discussed in section 2.3.1 has the advantage of also averaging the amount of warping between the two time series being averaged (Figure 3.3). However, these pairwise schemes do not minimize Eq. (2.6) as effectively as DBA [11].

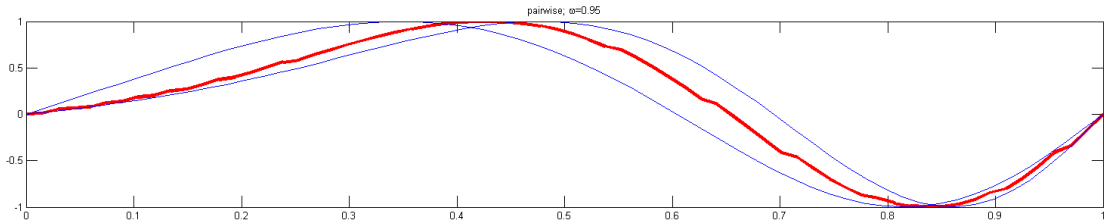


Figure 3.3: Average (red) of two time series (blue) by association

It would be preferable to have a method of averaging that can also average sequences

temporally, but still has the advantages of the low WGSS solution and order independence of DBA.

3.4.1 TPD with cooling parameter

To make DBA more robust to differences in time shifting, we instead use time penalized DTW from Section 3.3 when making barycenter associations. This forces spatial associations to compete with differences in time. The tradeoff is that making the time axis too stiff alters the shape of the average. In Fig. (3.4), three pulses were aligned using TPD with $\lambda = 1, 5$, and 10. With large values of λ, TPD does not want to warp the time axis much at all; this results in the peaks of the pulses not being aligned with one another, and consequently the average incorrectly has more than one peak.

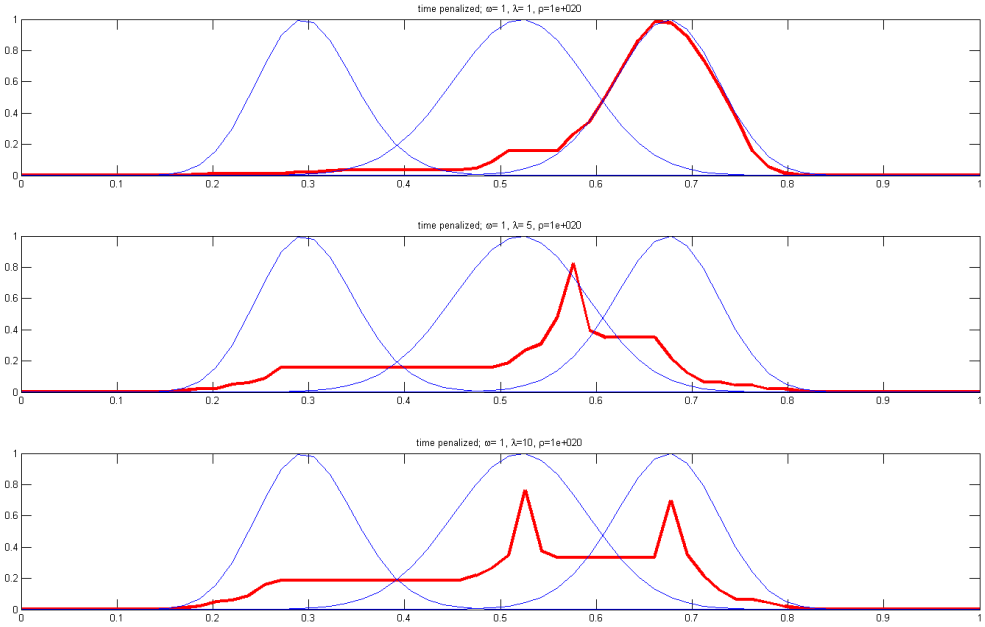


Figure 3.4: DBA using TPD with $\lambda = 1, 5$, and 10 respectively

We do not wish to use time dependence when calculating the final average, but using some time penalty initially will help pull meaningful features closer to the average time at which they happen. We also use relaxation to ensure that the time penalty does not interfere too much with the overall shape of the average.

At each iteration, the time penalty is calculated by

$$\lambda = \lambda_0 \left[1 - \left(\frac{q}{q_f} \right)^{\rho} \right] \tag{3.5}$$

where λ_0 is the initial penalty, q_f is the number of iterations, and ρ is a shape parameter (Fig. (3.5)). This has the effect of averaging the time series temporally before averaging them spatially.

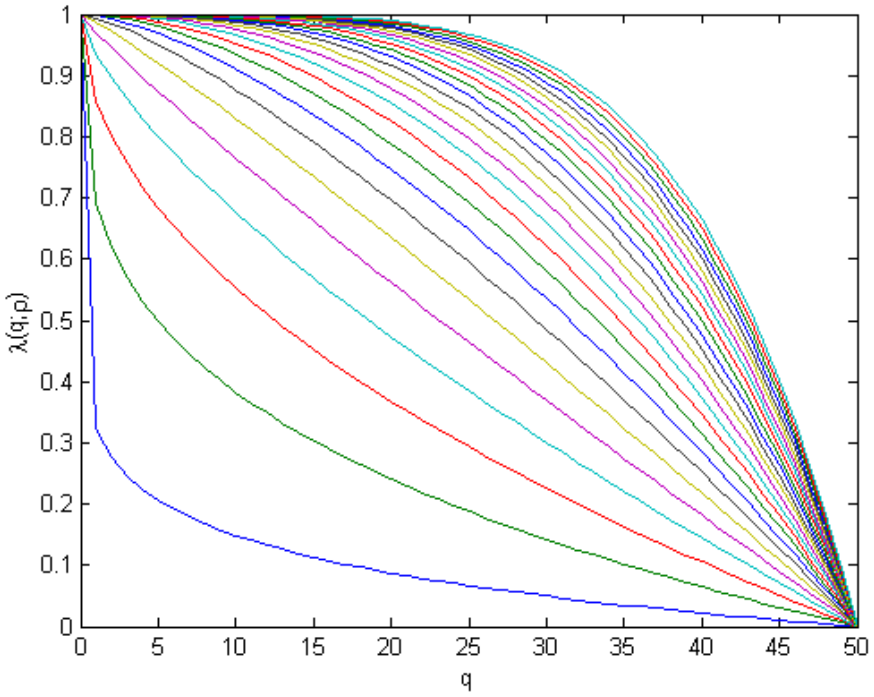


Figure 3.5: Proportion of λ_0 used at iteration q, for various values of ρ; curves toward the bottom correspond to smaller values of ρ.

Given this modification to classical dynamic time warping, our new algorithm is otherwise identical to that in [11], with the exception that instead of keeping C' as our new average, we instead keep $\omega C' + (1 - \omega)C$, where ω is the relaxation parameter, which was observed to aid convergence.

For completeness, the process in its entirety is outlined in Algorithm 3.1. We call it **TPD Barycenter Averaging** (**TBA**; cf. DTW Barycenter Averaging [11]). Observe that Eq. (3.5) has been modified so that κ iterations with no time penalty are applied at the end. The effect of the κ factor along with cooling λ is that the algorithm treats

time penalty as a preconditioner, rather than a means to the actual solution. We found $\kappa = 5$ works well.

Algorithm 3.1 TPD Barycenter Averaging with cooling and relaxation

for $q = 1 : q_f$ **do**
  $\kappa \leftarrow 5$;
  $\lambda \leftarrow \max\left\{0, \lambda_0 \left[1 - \left(\frac{q}{q_f - \kappa}\right)^{\rho}\right]\right\}$;
  $assoctab \leftarrow zeros(dim + 1, T_C)$
  for all $seq \in \mathbb{S}$ **do in parallel**
    $M \leftarrow DTW(C, seq; \hat{\delta}_\lambda)$;
    $(i, j) \leftarrow (T_C, T_{seq})$;
    while $i > 0$ & $j > 0$ **do**
      $assoctab(2 : end, i) \leftarrow assoctab(2 : end, i) + seq(:, j)$;
      $assoctab(1, i) \leftarrow assoctab(1, i) + 1$;
      $(i, j) \leftarrow M(i, j).predecessor$;
    end while
  end for
  $C' \leftarrow assoctab(2 : end, :)/assoctab(1, :)$;
  $C \leftarrow \omega C' + (1 - \omega)C$;
end for
return C

Cooling λ mitigates the problems in Fig. (3.4). As shown in Fig. (3.6), using time penalty pulls the peak toward the center temporally while still preserving the overall shape.

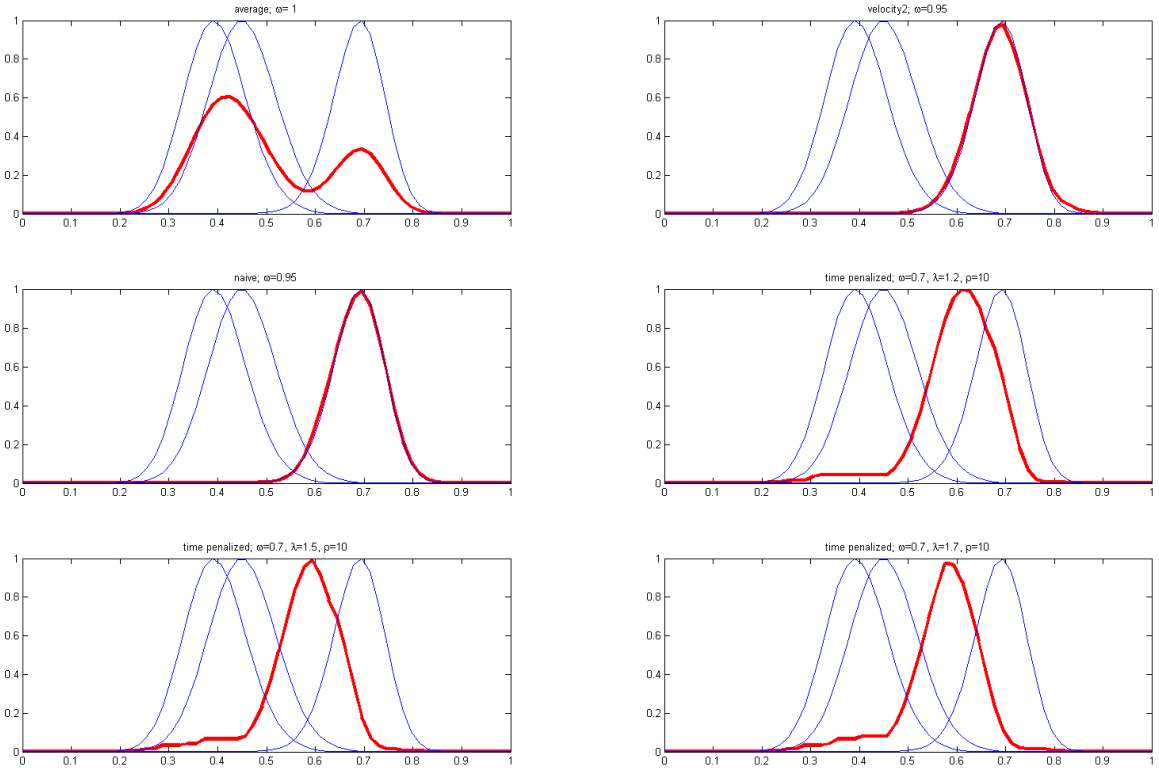


Figure 3.6: Top: left, arithmetic average of three pulses; right, average computed using DBA with centered difference DDTW. Middle: left, average computed using naive DTW; right, TPD with $\lambda_0 = 1.2$, $\rho = 10$. Bottom: left, TPD with $\lambda_0 = 1.5$, $\rho = 10$; right, TPD with $\lambda_0 = 1.7$, $\rho = 10$.

3.4.2 Effects of relaxation and shape parameter on contrived data

To provide warping to data, we compose the function with a time shift function

$$s(t; \mu, t_0) = t - t(t-1)\frac{\mu}{1 + (t - t_0)^4}. \tag{3.6}$$

We randomly generate values of t_0 and μ from uniform distributions; t_0 from $U[\frac{1}{12}, \frac{11}{12}]$ and μ from $U[-1, 1]$. Fig. (3.7) shows 80 of such warpings.

To investigate the effects of ρ and ω on the average, we generated five warpings of the Gaussian pulse $f(t) = e^{-(x-\frac{1}{2})^2/.01}$. We tested TBA with $\lambda = 2, 3$, and 5; $\rho = 1.5, 2, 7$, and 15; and $\omega = 0.6, 0.9$, and 1.2. As a measure of how "good" the average is spatially, we compare the resulting WGSS of the average, given by Eq. (2.6). A smaller value

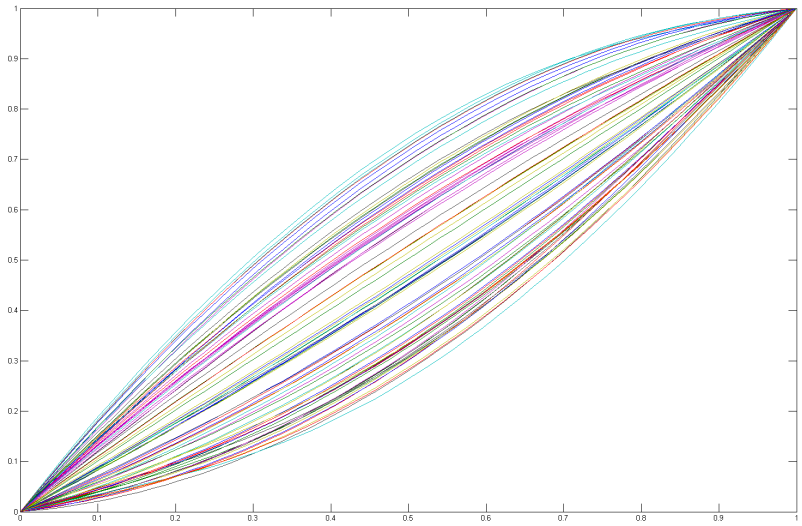


Figure 3.7: $s(t; \mu, t_0)$ for 80 randomly generated pairs μ, t_0.

for WGSS corresponds to a better approximation of the average. As a measure of how "good" the average is temporally, we visually compare the time coordinate of the peaks in each time series with that of the average.

All of these averages are shown in Figures 3.8, 3.9, 3.10, 3.11, and 3.12. Within each image, plots toward the bottom have a larger λ and plots toward the right have a larger ω. All trials were conducted using 35 iterations. The WGSS for each of these trials is summarized in Table 3.2.

For reference, relaxed DBA with no time penalty resulted in a WGSS of $1.41e-3$ for $\omega = 0.6, 0.7, 0.8, 0.9$, and 1; and $1.03e-3$ for $\omega = 1.1$ and 1.2.

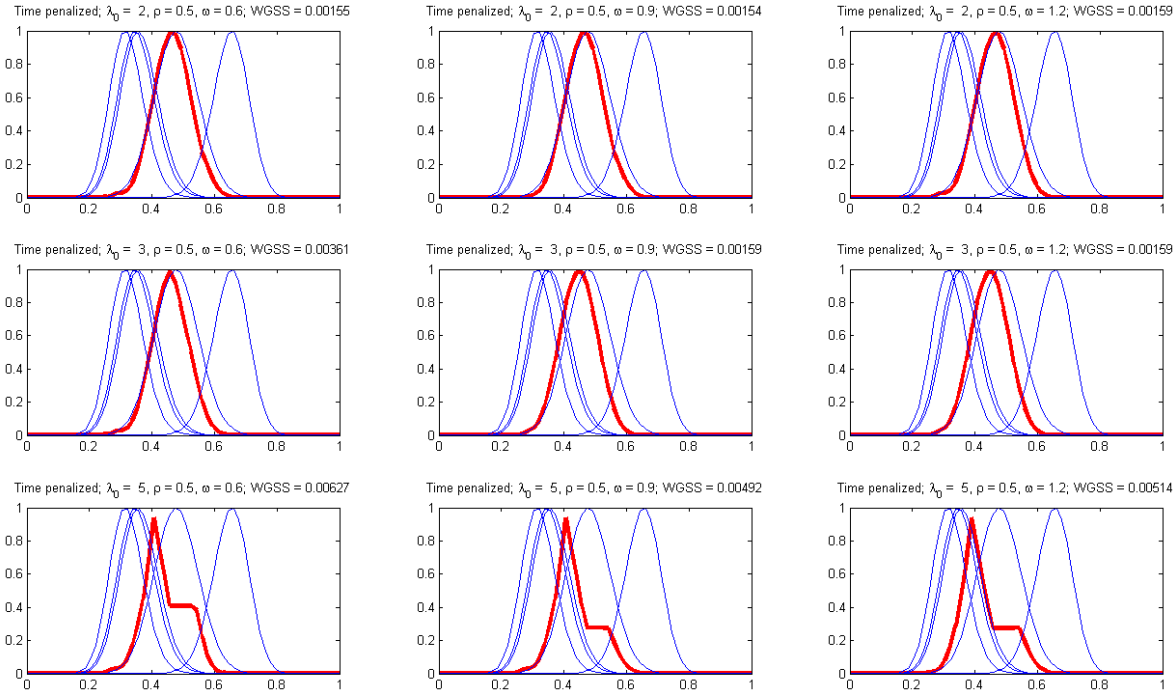


Figure 3.8: Varying λ and ω; $\rho = 0.5$

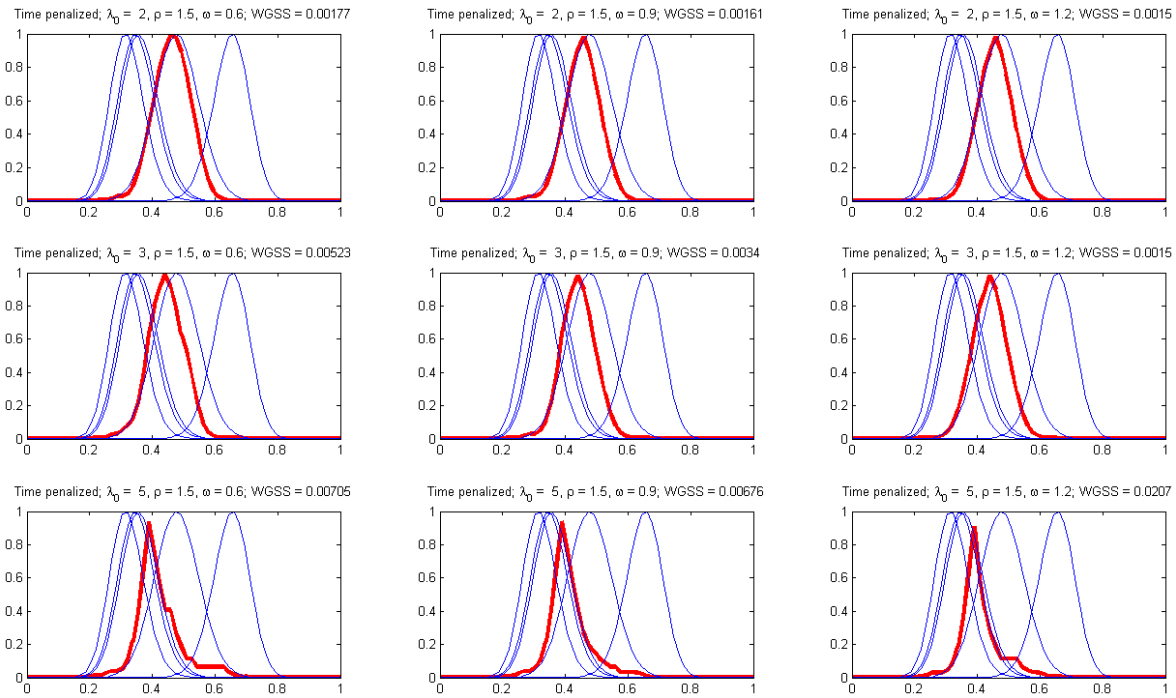


Figure 3.9: Varying λ and ω; $\rho = 1.5$

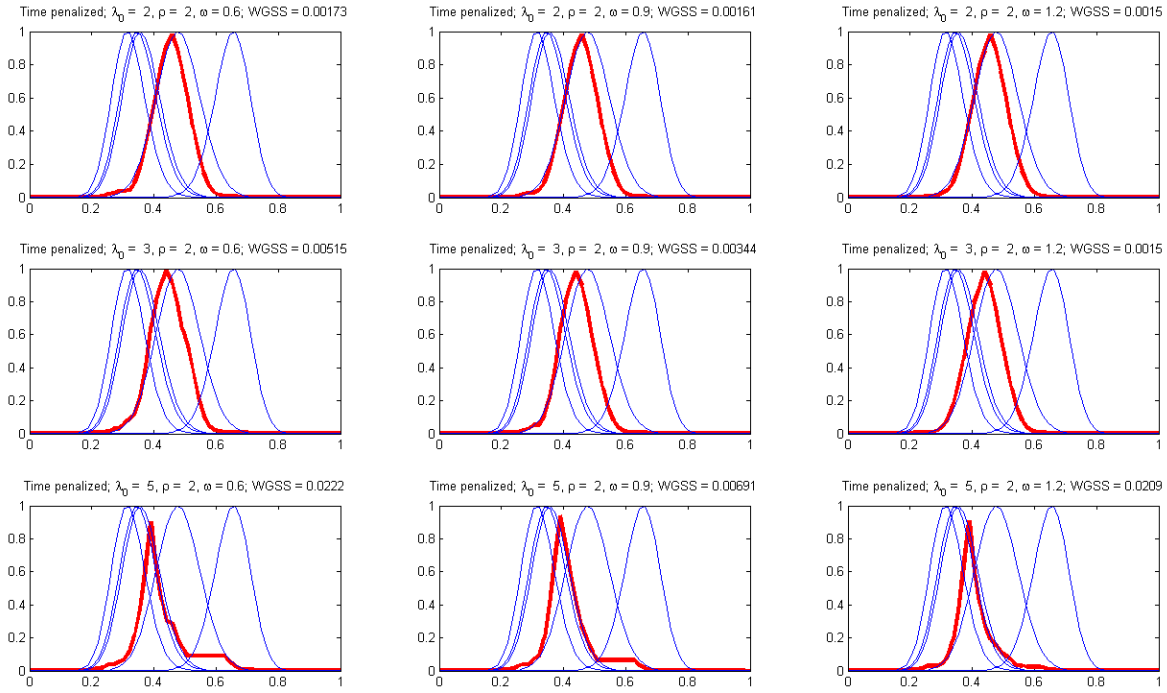


Figure 3.10: Varying λ and ω; $\rho = 2$

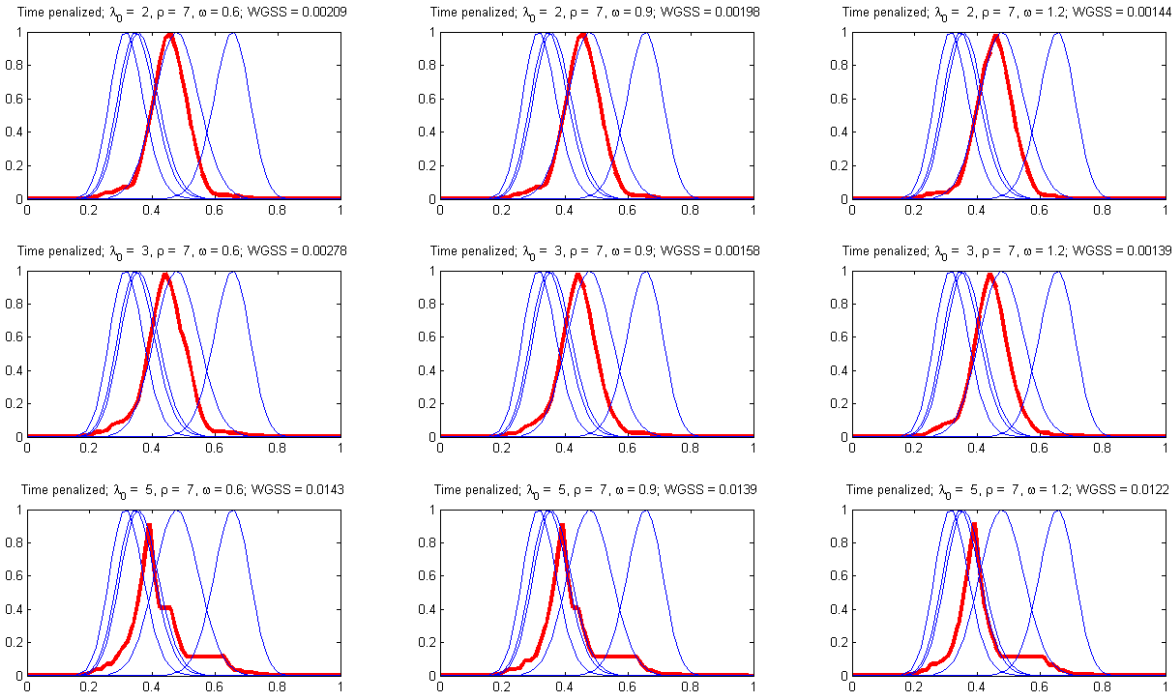


Figure 3.11: Varying λ and ω; $\rho = 7$

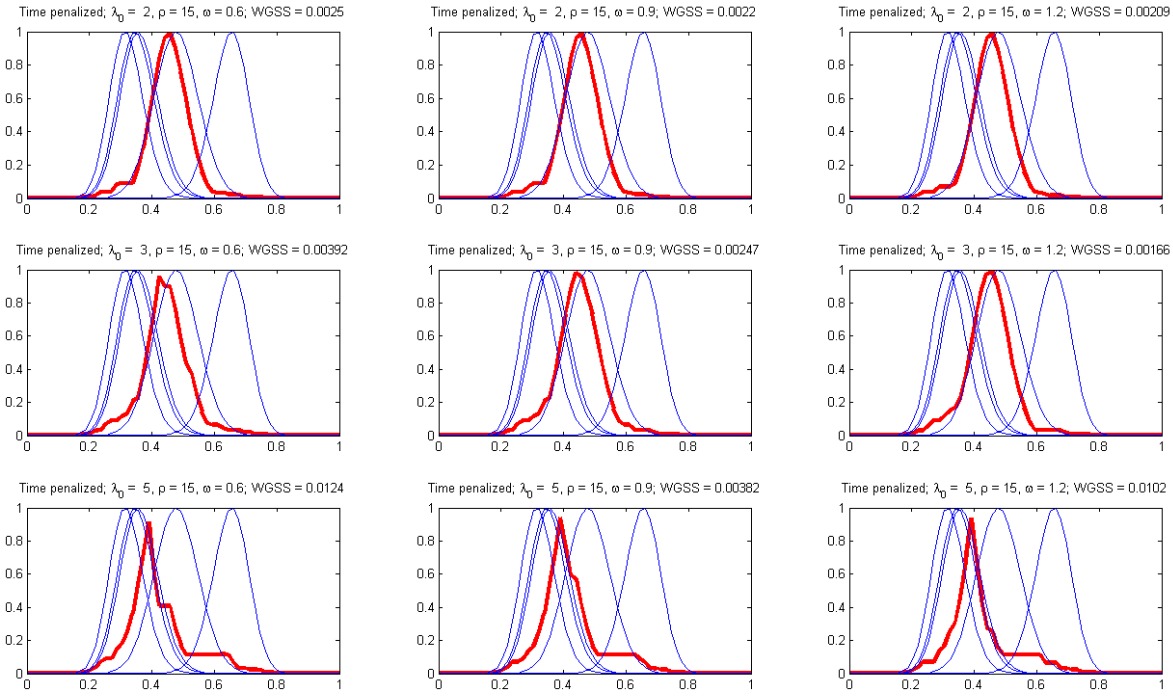


Figure 3.12: Varying λ and ω; $\rho = 15$

Table 3.2: WGSS for five Gaussian pulses using Time Penalized DBA

λ	ω	$\rho = 0.5$	$\rho = 1.5$	$\rho = 2$	$\rho = 7$	$\rho = 15$
2	0.6	1.55e-3	1.77e-3	1.73e-3	2.09e-3	2.50e-3
2	0.9	1.54e-3	1.61e-3	1.61e-3	1.98e-3	2.20e-3
2	1.2	1.59e-3	1.50e-3	1.50e-3	1.44e-3	2.09e-3
3	0.6	3.61e-3	5.23e-3	3.44e-3	2.78e-3	3.92e-3
3	0.9	1.59e-3	3.40e-3	5.15e-3	1.58e-3	2.47e-3
3	1.2	1.59e-3	1.50e-3	1.50e-3	1.39e-3	1.66e-3
5	0.6	6.27e-3	7.05e-3	2.22e-2	1.43e-2	1.24e-2
5	0.9	4.92e-3	6.76e-3	6.91e-3	1.39e-2	3.82e-3
5	1.2	5.14e-3	2.07e-2	2.09e-2	1.22e-2	1.02e-2

Discussion

Singularities result in "stalls" in the average. This happens when several coordinates in a row get updated to the same value, resulting in continuous sections of time along which the average is constant. Because DTW treats these essentially as a single point, they are not attenuated. TBA helps attenuate these stalls, but this is sensitive to the parameters.

Higher values of λ provide better temporal averaging but at the cost of distorting the result by introducing stalls. A small, but not excessively small ($\rho = 1.5$ in the tests above) value of ρ seems to be the best for preventing this if a large λ is used.

Too much underrelaxation causes suboptimal minima of WGSS as seen in Table 3.2; $\omega = 0.6$ almost universally results in a larger WGSS than does 0.9 except for a couple anomalous cases. However, $\omega = 1.2$ is too much in most cases, especially when λ is larger.

Larger values of λ typically perform better with smaller values of ρ, although this trend is not at all uniform. Smaller values of ρ allow more temporal flexibility in later iterations once initial temporal alignment has been attained.

3.4.3 Sensitivity to initial guess

Standard DBA is sensitive to the choice of initial guess; being more effective if the initial guess is one of the time series in the collection [11]. We would like to investigate how TBA performs under a variety of initial guesses. To do so, we are using the same five pulse functions as before, but with nine different initial guesses:

1. Four of the pulses in the collection;
2. $\sin(2\pi t)$;
3. A time series of all zeros;
4. A time series of random values from $U[0, 1]$;
5. t itself;
6. $t(1 - t)$.

In fact, TBA is not very particular about the initial guess, although picking one of the time series from $\mathbb{S}$ is typically more effective—but unlike classical DBA, TBA has excellent convergence with other initial guesses as well. In particular, a "model time series" exhibiting the theoretical shape of the gesture would work well as an initial guess.

In particular, the function $t(1-t)$ is a fairly rough model of the Gaussian pulses being tested, but in many cases TBA converges with this initial guess to a lower WGSS solution in the same number of iterations than some of the time series in $\mathbb{S}$.

We tested these initial guesses with a collection of 5 Gaussian pulses and with collections of 30 sinusoidal waveforms of varying frequencies. All trials for TBA used $\lambda = 1.5$ and $\rho = 5$.

Higher frequency data poses more of a problem for any algorithm due to the deterioration of resolution, but TBA minimizes WGSS better than classical DBA in all cases in which the initial guess is not chosen from $\mathbb{S}$, and in all but a few cases in which it is.

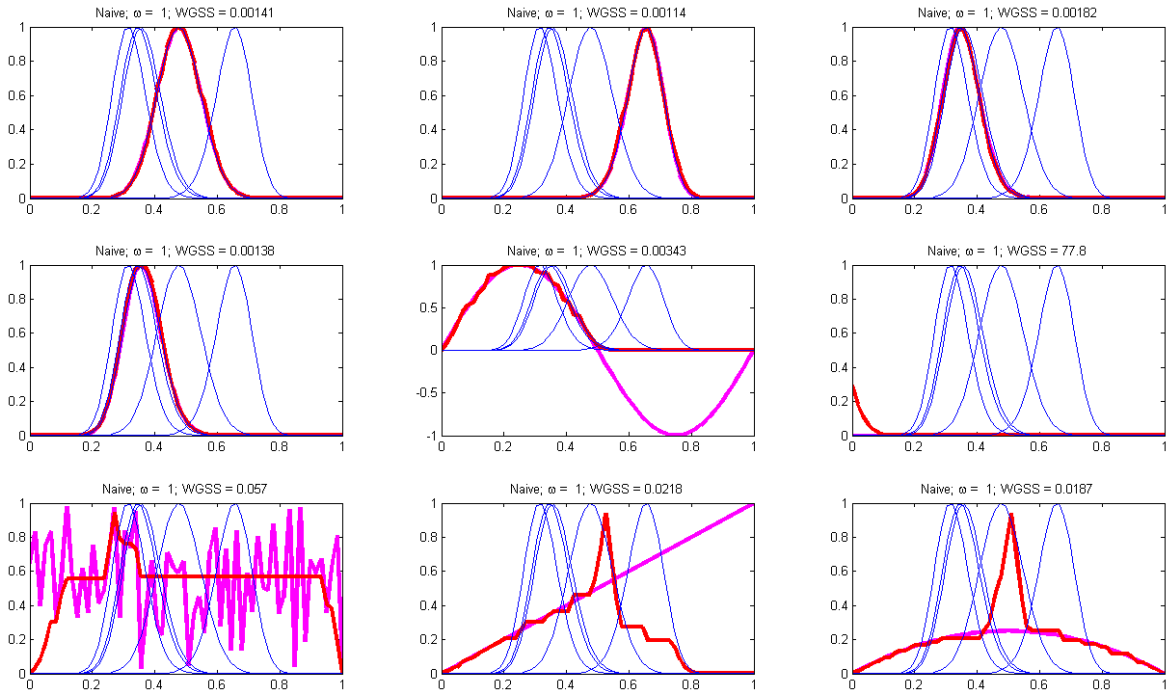


Figure 3.13: Various initial guesses (magenta) and the average (red) using naive DBA

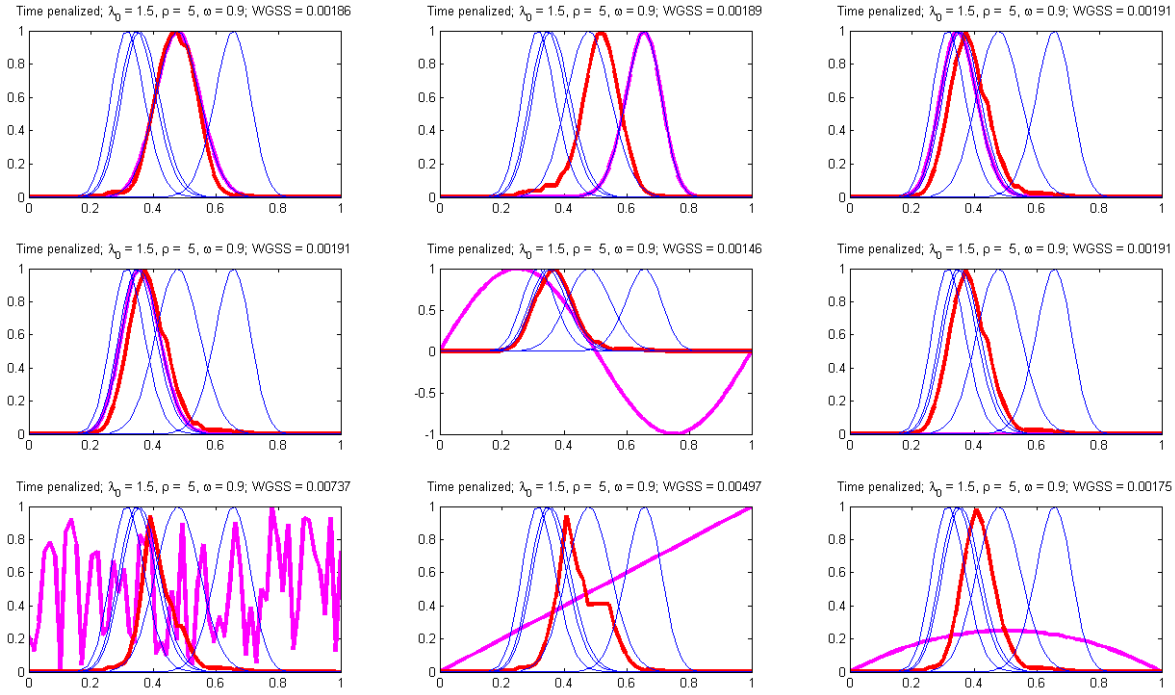


Figure 3.14: Various initial guesses (magenta) and the average (red) using TBA with $\omega = 0.9$

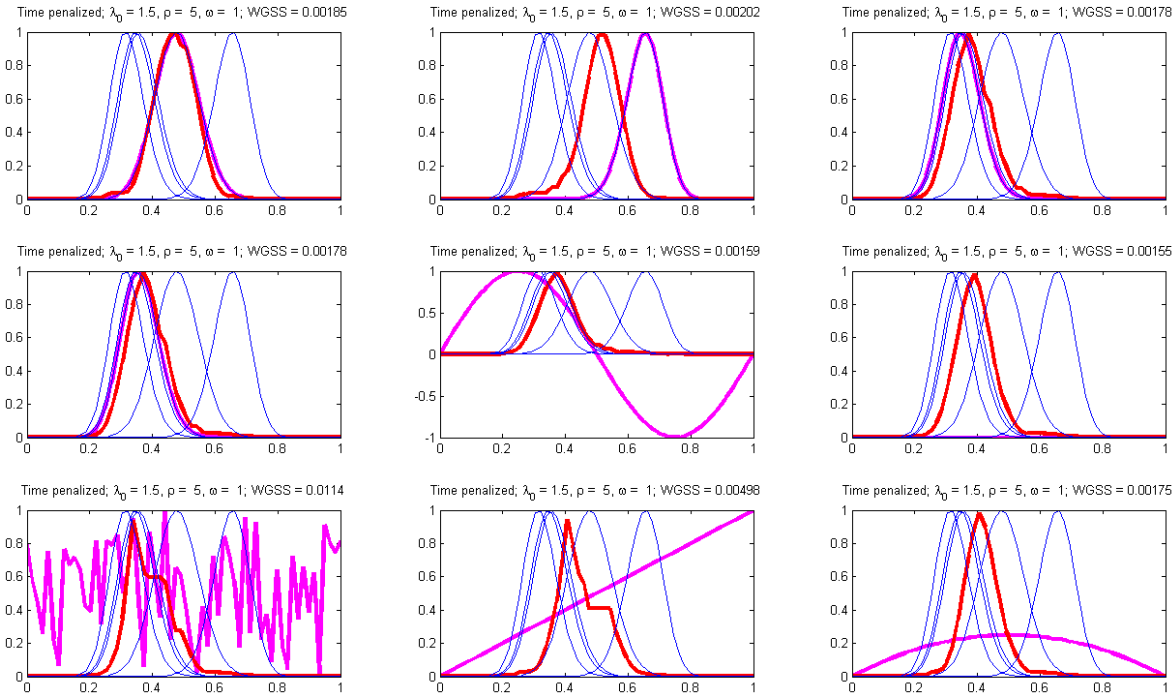


Figure 3.15: Various initial guesses (magenta) and the average (red) using TBA with $\omega = 1$

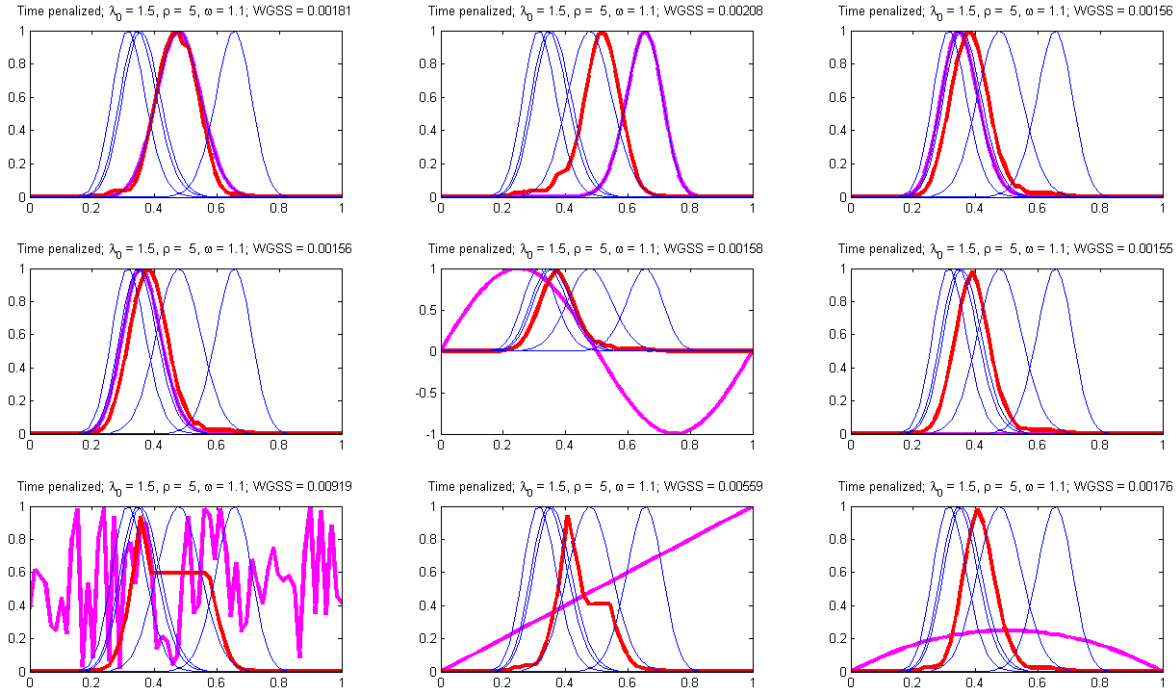


Figure 3.16: Various initial guesses (magenta) and the average (red) using TBA with $\omega = 1.1$

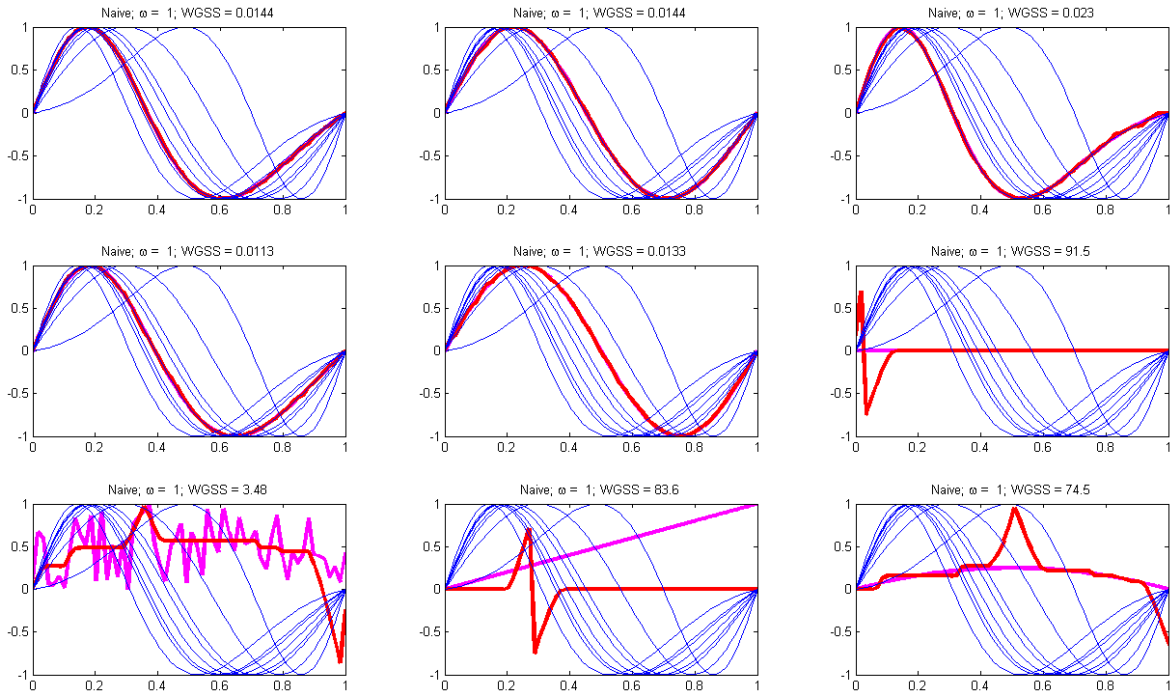


Figure 3.17: Various initial guesses (magenta) and the average (red) using naive DBA. Sequence is $\sin(2\pi t)$.

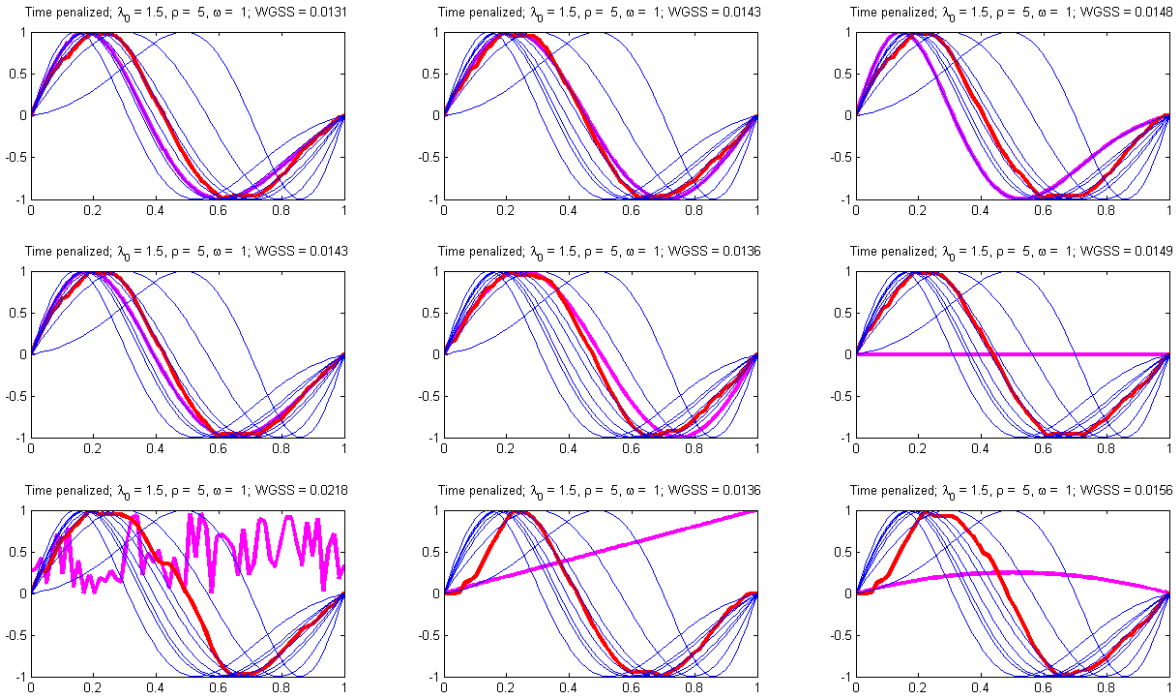


Figure 3.18: Various initial guesses (magenta) and the average (red) using TBA with $\omega = 1$. Sequence is $\sin(2\pi t)$.

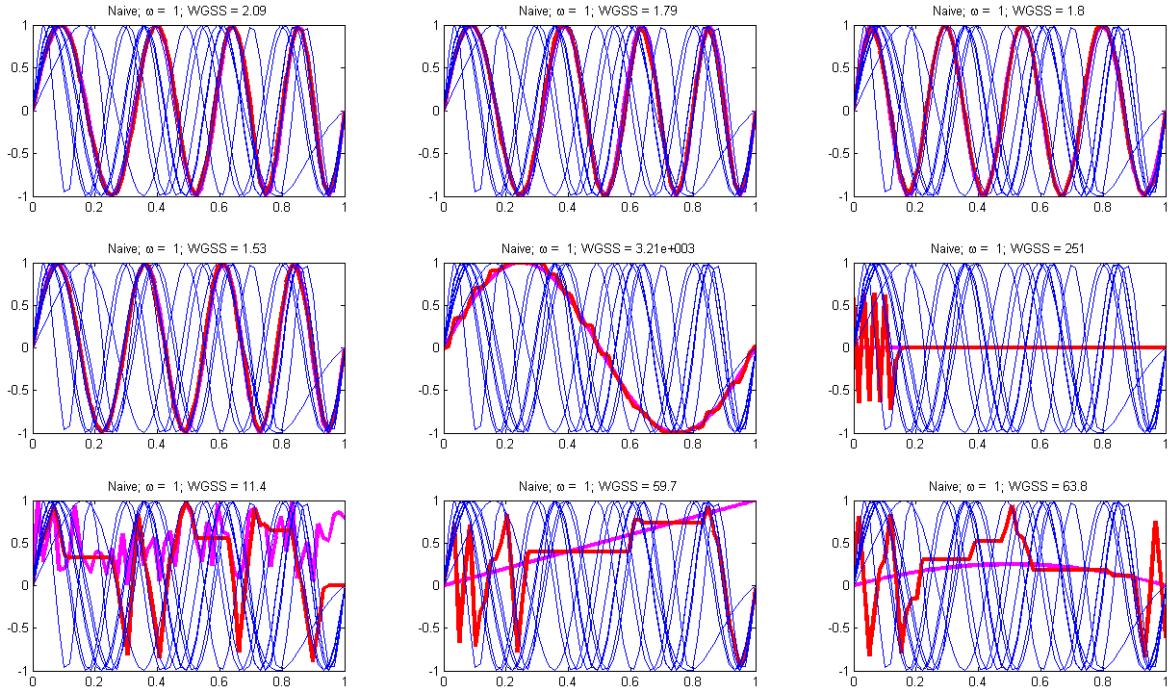


Figure 3.19: Various initial guesses (magenta) and the average (red) using naive DBA. Sequence is $\sin(4\pi t)$.

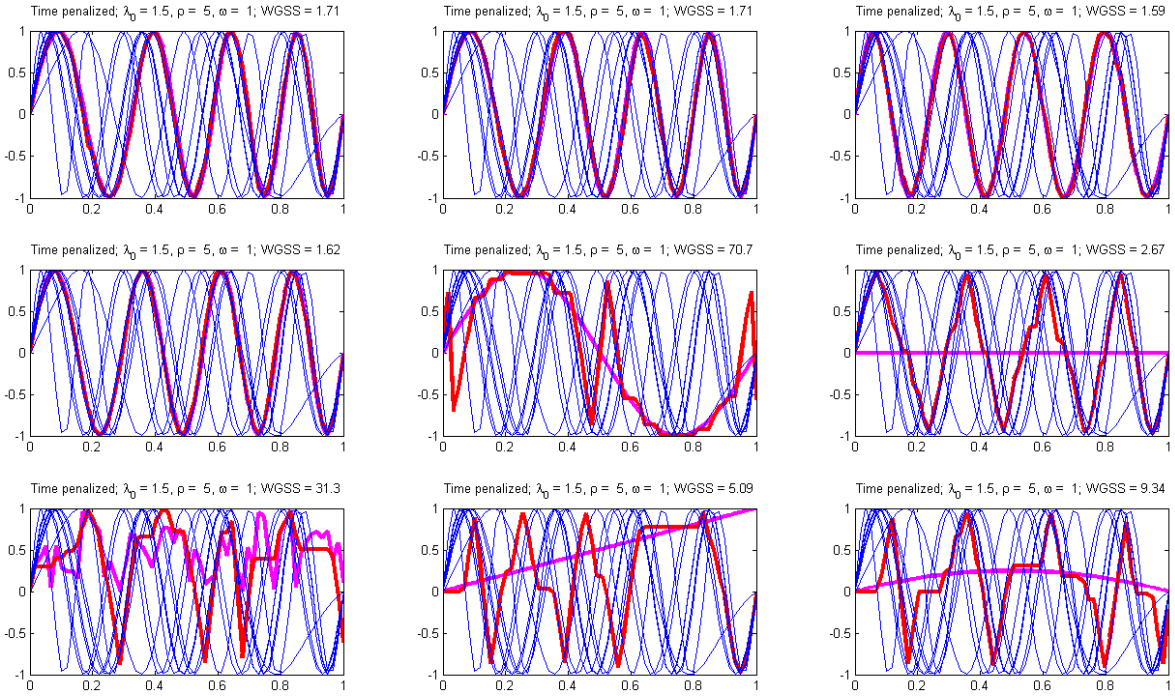


Figure 3.20: Various initial guesses (magenta) and the average (red) using TBA with $\omega = 1$. Sequence is $\sin(4\pi t)$.

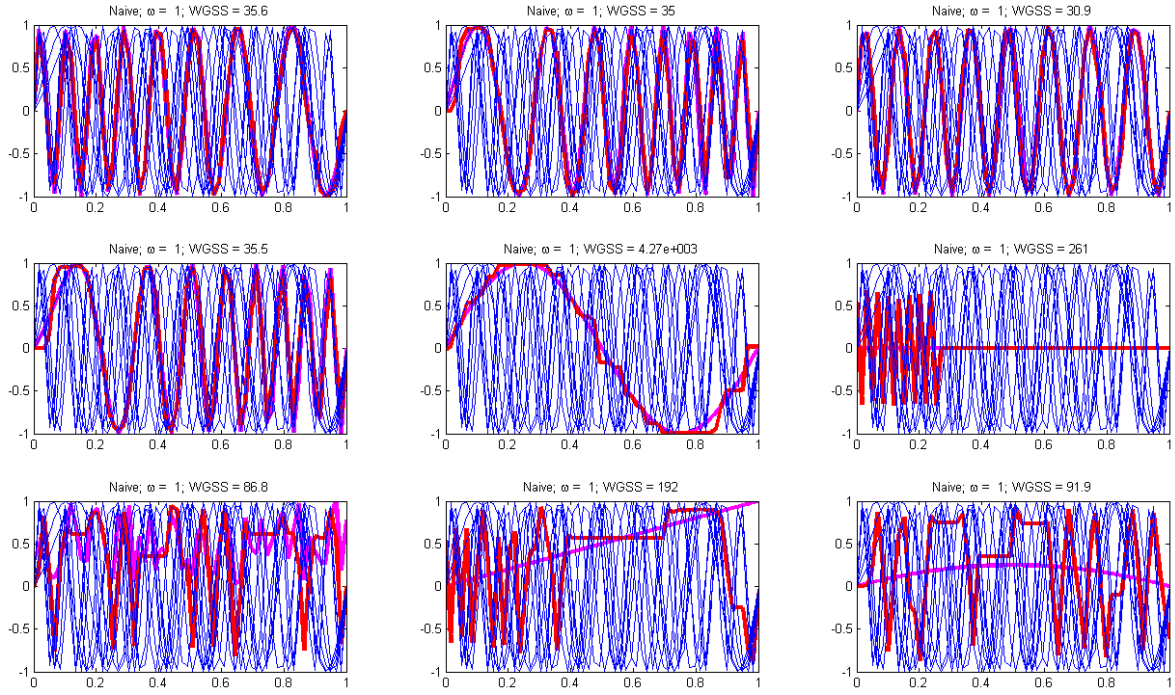


Figure 3.21: Various initial guesses (magenta) and the average (red) using naive DBA. Sequence is $\sin(16\pi t)$.

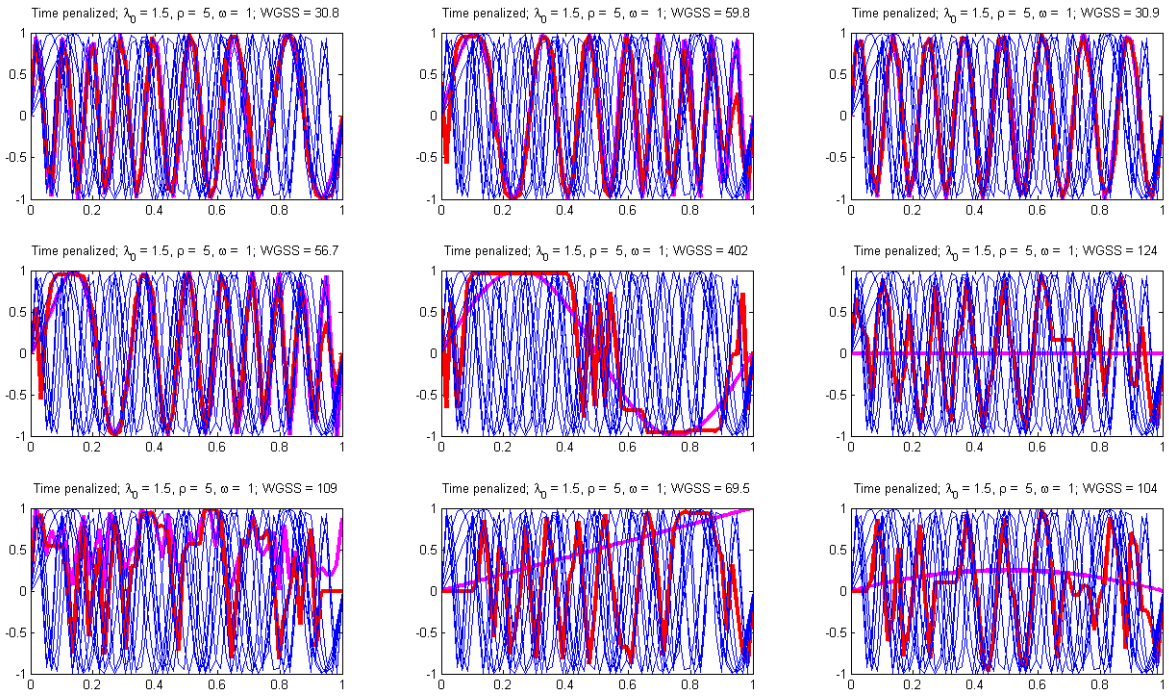


Figure 3.22: Various initial guesses (magenta) and the average (red) using TBA with $\omega = 1$. Sequence is $\sin(16\pi t)$.

Discussion

Time penalized DTW introduces commonality between time series; namely, a uniform time axis shared by all time series in the collection. This gives bad initial guesses a chance to make the correct associations with the time series in the collection. The common time axis gives DTW a "starting point" at finding the optimal warping path, rather than naively weighting any warping path the same. This is similar to windowing [3] but is far less strict.

Conversely, naive DTW treats all warping paths as equally likely candidates, which is too loose an assumption in time series collections that are not expected to have unbounded variation in their time scales. Time penalty allows a more precise average by encouraging warping paths to stay closer to the diagonal.

3.5 Summary

We have proposed a continuous formulation of the time series alignment problem. This formulation requires interpolation of the time series data, but relaxes the requirement that a point in one time series must be associated with a single point in the other.

We have also proposed two cost functions that can be used in dynamic time warping: the distance between the derivative using the centered difference approximation, and the time penalty cost function, which is a regularization of a standard cost function and is actually a family of cost functions.

We demonstrated a method of producing collections of artificial signals that are time warpings of each other, and showed how TBA produces an average for these collections in a manner that is better temporally aligned and less sensitive to initial guesses than DBA is.

Chapter 4

Experimentation on Data from Sign Language Videos

Chapter 3 introduced a modification to DTW Barycenter Averaging [11] called Time Penalized DTW Barycenter Averaging (TBA), and demonstrated its superior performance on artificially produced data sets.

We tested the algorithm on application-specific data as well. Our project constructed a sign language script and recorded 25 participants performing these signs on video using the Microsoft Kinect. For each video, we choose the (x, y, z) coordinates of the right palm as the frames of the time series corresponding to the video. These locations are provided by the OpenNI SDK [2] and no further image processing was done. For a selection of the signs recorded, we will show that TBA performs more favorably than classical DBA in averaging seven collections of signs exhibiting the same hand motion.

4.1 Dictionary & Testing Methodology

The vocabulary of signs recorded is summarized in Table 4.1. Many of these signs do not involve hand movement other than possibly wrist movement, however some broad classes of common movements are summarized in Table 4.2.

For experimentation purposes, we will select signs that are in both tables:

- *league* from the first gesture class;
- *maybe* from the second gesture class;
- *game* and *shoe* from the third gesture class;
- *committee* from the fourth gesture class;
- *lord* from the fifth gesture class;
- *please* from the sixth gesture class;
- *logic* from the seventh gesture class.

The signs in the third gesture class are placed in the same collection, as they involve the same gesture. We will compare the WGSS of each of these collections to the average found by:

1. Classical DBA using naive dynamic time warping (Classical DBA);
2. Classical DBA using derivative dynamic time warping with derivative estimate given by Eq. (2.5) (KP DDTW);
3. Classical DBA using derivative dynamic time warping with derivative estimate given by Eq. (3.3) (CD DDTW);
4. TBA with every combination of λ, ρ, and ω as $\lambda = 2, 3, 5$, $\rho = 0.5, 2, 7$ and $\omega = 0.9, 1.0, 1, 1$.

The hand location data from each individual sign was normalized to have a barycenter of zero and a standard deviation of one.

Figure 4.1: A signer performing the sign for "family".

Table 4.1: ASL Vocabulary included in database

Set 1	North	South	East	West
Set 2	please	thank you	maybe	late
Set 3	family	patience	always	true
Set 4	league	license	logic	lord
Set 5	yes	no	committee	senator
Set 6	what	America	wresting	class
Set 7	smart	apple	game	shoe
Set 8	mother	parent	grandmother	uncle

Table 4.2: Movement classes

Movement classification	Horiz. circle	up and down alt.	hitting hands horiz.	left shoulder to right shoulder	left shoulder to right hip	vert. circle on chest	head to neutral space	r. hand hitting l. hand	hand up and down on chest
One/two hands	two	two	two	one	one	one	one	two	one
	family	awkward	game	Toronto	princess	please	theory	approve	Albany
	department	diverse	shoe	Atlanta	lord	sorry	logic	park	Burlington
	organization	maybe	same	committee	king	disgusted	concept	stone	Chicago
	association	doubt	ball	senate	queen		idea	due	Detroit
	class	court	roommate	board			smart		Indianapolis
	group		license				brilliant		Philadelphia
	league		boot				hello		Rochester
	society								Texas
	team		football						Vancouver

4.2 Results & Discussion

The $\log(WGSS)$ for each class with each algorithm is summarized in Table 4.3. Note that each raw video originally contained four signs. To simplify the editing process, the videos were simply split into four quarters. A frame-by-frame analysis would have produced better data collections, but due to the volume of videos at our disposal, the time required to do so would have been impractical. Since DBA is already known to provide good results, experimentation is only concerned with comparable, rather than absolute performance.

Hence, the WGSS is fairly poor in all cases. However, it can still be seen that TPD barycenter averaging performs comparably to DTW barycenter averaging on the data available, showing that time penalty regularization is a suitable augmentation of DBA. Given the other benefits of TBA, we recommend exploring its use in means-based gesture recognition algorithms.

From each class, the result of the best performing algorithm is highlighted in yellow. For each gesture class except Class 6 (the word "please"), there exists a combination of λ, ρ, and ω for which TBA outperforms classical DBA. Derivative dynamic time

Table 4.3: $\log(WGSS)$ for each collection with various calculations of the average

Method	ω	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Classical DBA	0.9	5.7046	5.8793	5.6009	5.7354	5.6527	5.7551	5.7986
	1.0	5.7013	5.8722	5.6078	5.7406	5.6518	5.7528	5.7720
	1.1	5.6940	5.8387	5.6225	5.7382	5.6584	5.7362	5.7787
KP DDTW	0.9	6.0282	6.2177	5.7641	5.9786	5.8072	6.0855	6.0470
	1.0	6.0306	6.2548	5.7640	5.9695	5.8663	6.0915	6.0805
	1.1	6.0428	6.2635	5.7453	5.9703	5.8600	6.1351	6.0400
CD DDTW	0.9	5.7086	5.9420	5.5984	5.7468	5.7136	5.7563	5.8037
	1.0	5.7095	5.8846	5.6165	5.7362	5.7104	5.7438	5.8061
	1.1	5.7104	5.8789	5.5905	5.7327	5.7013	5.7556	5.7926

λ	ρ	ω							
2	0.5	0.9	5.6420	5.8336	5.5876	5.7257	5.6589	5.7798	5.7651
2	0.5	1.0	5.6433	5.7760	5.5944	5.7262	5.6566	5.7856	5.7346
2	0.5	1.1	5.6441	5.7779	5.6098	5.7294	5.6625	5.7754	5.7346
2	2.0	0.9	5.6760	5.7908	5.5945	5.7417	5.6662	5.7685	5.7218
2	2.0	1.0	5.6747	5.7656	5.6072	5.7441	5.6674	5.7744	5.7247
2	2.0	1.1	5.6517	5.7679	5.6128	5.7453	5.6665	5.7623	5.7261
2	7.0	0.9	5.6971	5.9486	5.6122	5.7149	5.6616	5.7872	5.7138
2	7.0	1.0	5.6397	5.9734	5.5858	5.7134	5.6588	5.7801	5.7053
2	7.0	1.1	5.6941	5.9700	5.5641	5.7207	5.6545	5.7804	5.7420
3	0.5	0.9	5.6815	5.7913	5.5480	5.7194	5.6600	5.7649	5.7616
3	0.5	1.0	5.6935	5.7793	5.5524	5.7328	5.6559	5.7620	5.7202
3	0.5	1.1	5.6474	5.8044	5.5596	5.7251	5.6516	5.7604	5.7298
3	2.0	0.9	5.7192	5.8365	5.5530	5.7254	5.6571	5.7698	5.7236
3	2.0	1.0	5.7242	5.8137	5.5608	5.7317	5.6566	5.7640	5.7186
3	2.0	1.1	5.6848	5.8043	5.5710	5.7338	5.6597	5.7627	5.7221
3	7.0	0.9	5.6254	5.9675	5.5678	5.7206	5.6863	5.7948	5.7050
3	7.0	1.0	5.6896	5.9740	5.5743	5.7173	5.6570	5.7901	5.7166
3	7.0	1.1	5.6887	5.9816	5.5441	5.7167	5.6601	5.7913	5.7140
5	0.5	0.9	5.7179	5.8326	5.5404	5.7504	5.6498	5.7545	5.7565
5	0.5	1.0	5.6627	5.8363	5.5492	5.7493	5.6570	5.7586	5.7225
5	0.5	1.1	5.6313	5.8776	5.5525	5.7464	5.6566	5.7515	5.7242
5	2.0	0.9	5.7297	5.9555	5.5567	5.7512	5.6694	5.7793	5.7264
5	2.0	1.0	5.7243	5.9497	5.5740	5.7376	5.6792	5.7507	5.7224
5	2.0	1.1	5.6489	5.9524	5.5799	5.7482	5.6651	5.7390	5.7338
5	7.0	0.9	5.6959	5.9792	5.5671	5.7201	5.6809	5.8001	5.7548
5	7.0	1.0	5.6502	5.9795	5.5458	5.7153	5.6818	5.7995	5.7324
5	7.0	1.1	5.6457	5.9708	5.5563	5.7156	5.6622	5.7993	5.7551

warping seems to provide no benefit in most cases, except in Class 3 (the words "game" and "shoe") where both the over- and under-relaxed trials with the centered difference version perform better than classical DBA.

As was found in Section 3.4, higher values of λ work better with smaller values of ρ and vice versa; the distinction being whether a high penalty is applied over a few iterations, or a moderate penalty is applied over many iterations.

A higher time penalty pulls relevant features such as peaks closer to the average time at which they happen across all time series being averaged. However, a larger time penalty also distorts the shape of the average.

The shape parameter also has a large effect. A larger shape parameter penalizes time over more iterations than a smaller one; however it does not appear to alter the amount of shifting. This may indicate that most of the shifting happens within the first couple of iterations.

The effects of relaxation are somewhat unpredictable. In some cases, changing ω reduces stalling (see Section 3.4), but there may be a more reliable way to accomplish this.

A major advantage of TBA is that it is less sensitive to the initial guess. This allows for flexibility in choosing the initial guess; in particular, a model trajectory can be artificially constructed and then refined.

4.3 Summary

We have tested TBA alongside classical DBA and DBA using derivative dynamic time warping and showed that TBA is an effective averaging scheme for the time series of hand location data from captured depth video. Moreover, the average found by TBA has a more uniform time scale than that given by classical DBA.

In addition to the benefit of having a more uniform time scale, TBA minimizes WGSS at least as effectively as DBA on real world data, making time penalty a suitable augmentation to this averaging scheme. Due to the benefits of TBA as explored in Chapter 3, we recommend exploring its use in means-based gesture recognition algorithms.

Chapter 5

Conclusions

We have justified the need for a method of averaging time series in gesture classification. We have also demonstrated how gesture classification can be applied to accessibility and education technologies for the deaf.

We have reviewed DTW Barycenter Averaging, a method of averaging a collection in a manner that is independent of the order of the collection. DBA produces higher quality averages than traditional pairwise methods, and is commutative and associative.

We then presented an improvement to DBA and showed on experimental data that it gives more consistent results regardless of the initial guess. Further, it is more effective at averaging the time at which events happen, and not just the events themselves. This prevents outliers from affecting the performance of clustering schemes that rely on a high quality average.

Future research may include other cooling schemes such as alternating high and low time penalties. Intuitively this may allow more prolonged application of time penalty while correcting the spatial errors it introduces in between. It is also possible to apply the time penalty regularization to other cost functions beyond the Euclidean distance; for instance, time penalty can be applied to the distance between the local derivatives of time series or to the hamming distance.

Given the proposal of our improvements to DBA, we recommend further investigation into its performance in means-based gesture recognition algorithms.

Bibliography

[1] Kinect for Windows: Voice, movement & gesture recognition technology. `http://www.microsoft.com/en-us/kinectforwindows/develop/`.

[2] Open–source SDK for 3D sensors—OpenNI. `http://www.openni.org/`.

[3] DJ Berndt and J Clifford. Dj berndt. In *J. Clifford. Using dynamic time warping to find patterns in time series. AAAI-94 Workshop on Knowledge Discovery in Databases (KDD-94)*, pages 359–370, 1994.

[4] Melissa M Burton, Chad Harbig, Mariam Melkumyan, Lei Zhang, and Jiyoung Choi. An evaluation of signbright: A storytelling application for sign language acquisition and interpersonal bonding amongst deaf and hard of hearing youth and caregivers. In *HCI International 2011–Posters Extended Abstracts*, pages 474–478. Springer, 2011.

[5] Jin-Woo Chung, Ho-Joon Lee, and Jong C Park. Improving accessibility to web documents for the aurally challenged with sign language animation. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, page 33. ACM, 2011.

[6] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[7] Matt Huenerfauth. Generating american sign language animation: overcoming misconceptions and technical challenges. *Universal Access in the Information Society*, 6(4):419–434, 2008.

[8] Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In *the 1st SIAM Int. Conf. on Data Mining (SDM-2001), Chicago, IL, USA*, 2001.

[9] Joseph B Kruskal and Mark Liberman. The symmetric time-warping problem: from continuous to discrete. *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 125–161, 1983.

[10] Evguenia Malaia, John Borneman, and Ronnie B Wilbur. Analysis of asl motion capture data towards identification of verb type. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 155–164. Association for Computational Linguistics, 2008.

[11] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[12] Hiroaki Sakoe and Seibi Chiba. A dynamic programming approach to continuous speech recognition. In *Proceedings of the Seventh International Congress on Acoustics*, volume 3, pages 65–69, 1971.

[13] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 26(1):43–49, 1978.

[14] Richard A Tennant and Marianne Gluszak Brown. *The American sign language handshape dictionary*. Gallaudet University Press, Washington, DC, 2 edition, 2010.

[15] Clayton Valli. *Linguistics of American sign language: An introduction*. Gallaudet University Press, 2000.

[16] Jie Yang and Yangsheng Xu. Hidden markov model for gesture recognition. Technical report, DTIC Document, 1994.

[17] Zahoor Zafrulla, Helene Brashear, Pei Yin, Peter Presti, Thad Starner, and Harley Hamilton. American sign language phrase verification in an educational game for deaf children. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3846–3849. IEEE, 2010.