Improving The Performance Of Similarity Metrics Used For Comparing Time Series Sequences

M. Adnan Haider

Master of Science
School of Informatics
University of Edinburgh

2013

Abstract

Acknowledgements

Many thanks to my mummy for the numerous packed lunches; and of course to Igor, my faithful lab assistant.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*M. Adnan Haider*)

Table of Contents

Chapter 1

Introduction

Over the course of the last decade, the mining of time-series data has received considerable attention within the data mining and machine learning community. The term 'time series' represents a sequence of numerical data points collected at regular intervals over a period of time. The duration of time period may be either in the order of milliseconds or monthly or even annually depending on the domain.

Mathematically, a time series is defined by the values $y_1, y_2...y_n$ at times $t_1, t_2...t_n$ where $y = f(t)$. The time t_i acts as an independent variable to estimate dependent variables y_i. The dimensionality of the series is denoted as **n** where the value of '**n**' is equal to the length of the sequence.

Time series analysis is used in many applications ranging from sales forecasting, budgetary analysis to stock market analysis and many more. One particular domain where the application of time series analysis is currently very popular is *motif* discovery- the problem of efficiently locating frequent/interesting sub-patterns in the data. The knowledge of motifs has been seen to have important applications in various aspects of data mining tasks. For instance motifs can use applied :

- to discover association rules [1] that the reflect information of 'primitive shapes.
- to specify the number of clusters for unsupervised clustering algorithms. The knowledge of motifs give a good approximation on the number of meaningful groups that are present in the data[2].
- to identify important sub-patterns in DNA and gene sequences [3].

In the analysis of speech data, motifs also play a very important role. Recent research have shown that detecting and isolating motifs in speech utterances is equivalent to identifying words that are specific to genres or in the case of recordings of individual speakers, detecting words that are frequently spoken by them. [4, 5].

To identify and extract motifs from time series data, various metrics have been proposed to match similar time series sequences. Among them, the most widely used is the Dynamic time warping algorithm(DTW) [6, 7, 8, 9, 10, 11] which matches similar sequences separated by time shifts, length or scale. By warping the time axis, the DTW algorithm compares each point in one sequence with points at different temporal regions in the second sequence. This allows the algorithm to be invariant to changes of speed, length or scale between similar patterns and thus provide a much richer comparison between time series sequences in comparison to standard metrics such as the Euclidean distance . Metrics such as the Euclidean distance are constrained to performing only linear matches between the temporal dimensions of sequences and require the all sequences to share the same length. This provides a limitation on the type of time series domains such metrics can be applied to. The speech corpus is an prime example of one such domain where distance metrics such as the Euclidean cannot be applied directly. In such data sets, recorded utterances of the same same lexical identity may not necessarily share the same dimensionality (i.e the same length) and thus may be contracted/expanded versions of each other due to speaker variations, context etc.

However, the DTW algorithm also experiences from severe drawbacks. The algorithm suffers from large run-times when the length of the time series sequences are very long. This is because the time complexity of the algorithm is quadratic i.e O(n^2) where n is the length of the sequence. In comparison, standard metrics like the euclidean has a time complexity of O(n), making them computational less expensive to employ. To address the quadratic time complexity of the DTW algorithm window constraints (Itakura parallelogram[12], Sakoe-Chiba band[7]) have been proposed to reduce the size of the search space of DTW by forcing the algorithm to look for optimal paths that are along the diagonal. Although introducing such window constraints does improve the time complexity, the reduction in the search space however leads to a substantial decrease in the accuracy of the algorithm since all optimal paths may not exist around the diagonal region[11].

The goal of this project is to employ machine learning techniques to tackle and resolve the individual drawbacks associated with using the DTW algorithm and the euclidean metric to compute the similarity between high dimensional time series sequences. In the first half of the project, I will be solely concentrating on improving the performance of the DTW algorithm for problem domains where the minimising the time complexity is as high priority as achieving good accuracy. In the latter half of the project, I will be concentrating on improving the performance of the euclidean metric in time series domains where intra class sequences share the same global shape. To be precise, I will be exploring feature extraction techniques to extract a smaller set of independent features from the raw sequences that can capture information about the global shape and be used by the euclidean metric to cluster/classify intra class sequences that share global trends.

For this project, I will be primarily using 3 time series datasets (details given in chapter 2):

- TIDIGITS corpus
- InLineSkate
- Cinc_ECG_TORSO

To evaluate and compare the merits of different proposed changes, I have partitioned each of the above datasets into a training set and a test set and employed the 1 nearest neighbour classifier to classify the test instances. In the first half of the project, the DTW was used as the similarity metric while for the later analysis, the euclidean metric was employed as the similarity metric. The performance of the classifier has been used to compare different proposed adaptions of the DTW and evaluate the performance of using different feature sets when employing the euclidean metric. The reason for choosing the 1 nearest neighbour classifier is that it shares almost the same methodology as the algorithms used to detect motifs. Motif detection algorithms are memory based i.e like 1NN classifier, they rely on comparing each sequence with other sequences in the dataset. Hence, methodologies that tend to achieve good performance on 1NN classification can be expected to achieve equivalent performance in motif detection.

The dissertation is organised as follows: Chapter 2 gives a brief description of the time-series datasets used for this project. Chapter 4 provides a detailed background description of the DTW algorithm. Chapter 3 and 4 investigates

methods to improve the performance of the DTW algorithm in terms of both accuracy and speed. Chapter 5 explores the novel technique of single value decomposition in the context of time series domains and explores whether it can be applied to extract a smaller set of latent features that can minimise the run time of the 1NN classifier when using the euclidean metric to classify sequences. Chapter 6 focusses on improving the accuracy of the classification of 1NN classifier on time series sequences when it employes the euclidean metric. In this chapter, I mainly explore preprocessing techniques that can extract novel features which capture the global properties of sequences. Finally in the last chapter, I present my conclusions and present scope for future work.

Chapter 2

Datasets Used

There are two primary goals of this project :

- improve the performance of the DTW algorithm on long time series sequences.
- explore the use of feature extraction methods to generate a small set of descriptive latent features[13] that can be used by the euclidean metric to cluster intra class sequences that share global trends more accurately.

The performance of the 1 NN classifier has been used to compare different proposed adaptions of the DTW and evaluate the performance of using different feature sets when employing the euclidean metric. For this project, the datasets used for evaluation were carefully chosen so that they contain time series sequences that have very **high** dimensions i.e long lengths.

The datasets used for this project are as follows:

1. TIDIGITS speech corpus
2. The UCR data set :CinC_ECG_torso
3. The UCR data set :InlineSkate

All 3 data sets were partitioned into two disjoints sets: test set and training sets. Each sample in the training and test data set corresponds to a particular pattern. The class information of the training samples were used to evaluate the accuracy of the nearest neighbours classifier.

For the first half of the project, the primary dataset that was mostly used is the 'TIDIGITS' corpus. In comparison to the sequences in the two UCR datasets,

the length of the time series sequences in the TIDIGITS corpus are on average 100 times longer. This forces the DTW algorithm to suffer from high time complexity when handling the speech utterances. The TIDIGITS corpus is a prime example where the minimisation of the run time is as high priority as getting good accuracy. In chapter 4 and 5, I will be investigating domain dependent and independent methodologies to improve the performance of the DTW in high dimensional time series datasets such as the TIDIGITS corpus.

The TIDIGITS corpus consists of time series sequences that are of variable length. Since the application of the euclidean metric is restricted to time series sequences that share the same length, for the later half of the project, I will be focussing mainly on the UCR datasets when exploring the use of various feature extraction methods that can improve the accuracy of the Euclidean metric in computing the similarity between intra class sequences that share the same length.

Description of the data sets is given below:

2.1 Brief description of TIDIGITS dataset

The TIDIGITS dataset is a large speech database that has been collected for the use in designing and evaluating algorithms for speaker-independent recognition of connected digit sequences. This dialectically balanced database consists of more than 25 thousand digit sequences spoken by over 300 men, women, and children. The data were collected in a quiet environment and digitised at 20 kHz[14]. The vocabulary of the corpus consists of :

- 22 isolated digits (two tokens of each of the eleven digits)
- 11 two-digit sequences
- 11 three-digit sequences
- 11 four-digit sequences
- 11 five-digit sequences
- 11 seven-digit sequences

In this corpus, the motifs correspond to individual digits. Thus for my experiments, I have considered only the isolated digit sequences for classification,

more specifically the digits from 0 to 9. The database was divided into two subsets, one to be used for algorithm design and one to be used only for evaluation. The division was based on speaker category $\{men, women, boy, girl\}$ and dialect classification, and yielded two sets of speakers, each containing approximately half the speakers of each category.

The table below shows the partitioning of speakers from different speaker categories in the training and test dataset.:

Subset	Man	Woman	Boy	Girl
Train	55	57	25	26
Test	56	57	25	25

For this project, only samples from one production have been used. To perform 1 NN classification, the entire training set have been used . However, to ensure that the experiments completed with in a reasonable time, I have only used $\frac{1}{3}$ of the samples of the test set.

The test data subset used for evaluation consists of

- 162 random samples from boy category
- 162 random samples from girl category
- 326 random samples from men category
- 326 random samples from women category

Note: The samples were not chosen exactly randomly. When subsampling from each category, I have ensured that:

- the test set contained samples from a rich variety of different speakers.
- the chosen subset consists of enough examples of all 10 digits.

2.2 Brief description CinC_ECG_torso and InLineSkate dataset

The UCR data resource has been funded by an NSF(National Science Foundation) since 2003 and is the largest public collection of time series data sets that

have been made available to the data mining/machine learning community, to encourage reproducible research for time series classification and clustering.

The InLineSkate dataset represents the EMG data of 7 major leg extensor and flexor muscles of the right side of the body from six athletes collected by attaching bipolar surface electrodes to the relevant muscles[15].The sampling rate was 1 kHz. The Cinc_ECG_Torso presents the ECG data collected by attaching electrodes to 4 distinct regions on the torso from multiple participants.

InLineSkate:

- Length of the time series:1882
- Size of test set:550
- Size of training set:100
- Number of classes:7

CinC_ECG_torso:

- Length of the time series:1639
- Size of test set:1380
- Size of training set:40
- Number of classes:4

2.3 Evaluation

The focus of this project is investigate methods to improve the performance of the DTW and the Euclidean metric in time series problem domains where minimising the time complexity is as important as improving accuracy. To evaluate and compare the merits of different proposed methods to improve the accuracy and the computational cost in comparing time series sequences when using the DTW or the Euclidean distance, the accuracy and time incurred by the 1NN classifier to classify all the instances in the respective test sets has been used as an evaluation criteria. As I have already mentioned in the previous chapter, motif detection algorithms like 1 NN are memory based i.e each sequence is compared with all sequences in the dataset. Thus, methods that improve the performance of the DTW and the Euclidean metrics in classifying sequences that have high dimensionality can be expected to improve the

performance of these metrics in detecting motifs in high dimensional time series datasets.

Chapter 3

Background

3.1 Dynamic Time Warping Algorithm

The Dynamic Time Warping algorithm is a distance metric that computes the similarity between time series sequences varying in length, scale and speed. The problem formulation [6] of the algorithm is stated as follows: Given two time series X, and Y, of lengths $|X|$ and $|Y|$,

$$X = x_1, x_2 ... x_{|X|} \tag{3.1}$$

$$Y = y_1, y_2 ... y_{|Y|} \tag{3.2}$$

construct a warping path W

$$W = w_1, w_2 ... w_k \text{ where max } (|X|, |Y|) \leq k \leq |X| + |Y|$$

where

- k denotes the length of the warping path
- the mth element of the warping path is $w_m = (i, j) \in [1 : N] \times [1 : M]$ for $l \in [1 : k]$ where i is an index from the time series X and j is an index from the time series Y.

To properly understand the above formulation some key definitions must be stated:

1. Warping path [16]: An (N,M)-warping path (or simply referred to as warping path if N and M are clear from the context) is a sequence

$w = (w_1, ..., w_k)$ with $w_l = (i, j) \in [1 : N] \times [1 : M]$ for $l \in [1 : k]$ that satisfies the following three conditions.

(a) Boundary condition [16, 7, 10, 12]: $w_1 = (1,1)$ and $w_k = (N, M)$.

The boundary condition enforces that the first elements of the series X and Y as well as the last elements of the series X and Y have to be aligned with each other. This prevents the alignment from missing any points.

(b) Monotonicity condition [16, 7, 12]:

$\forall i \in [1 : k-1], [|w_{i+1} - w_i| = 1]$. In other words, $w_k = (i, j), w_{k+1} = (i', j'), i \leq i' \leq i+1, j \leq j' \leq j+1$

The condition requires that the path will not turn back on itself. Both the i and j indexes either stay the same or increase but they will never decrease.

(c) Step-size condition [16, 7, 12]: $w_{l+1} - w_l \in \{(1,0), (0,1), (1,1)\}$, $\forall l \in$ [1:k-1].

The step size condition expresses a kind of continuity condition: no element in X and Y can be omitted and there are no replications in the alignment

Intuitively speaking, the (N,M) warping path $w = (w_1, ..., w_k)$ defines an alignment between two sequences $X = (x_1, x_2, ..., x_N)$ and $Y = (y_1, y_2, ..., y_M)$ by assigning each element of X to a unique element in Y.

2. Optimum Warping Path [10, 16]:

The optimal warp path corresponds to the minimum-distance warp path, where the distance of a warp path W is given as

$$Dist(W) = \sum_{m=1}^{K} dist(X, Y)_{|(w_m)}$$

where $w_m = (i, j) \in [1 : N] \times [1 : M]$

The $dist(X, Y)_{|(w_m)}$ represents the distance computed using an appropriate cost function between the value x_i of sequence X and the value y_j of sequence Y.

$$dist(X, Y)_{|(w_m)} = dist(x_{x_i}, y_{y_j})$$

The goal of the DTW algorithm is to compute the distance of the optimal warping path for given two time series sequences. Instead of attempting to solve the entire problem all at once, the algorithm utilises the technique of dynamic programming to find an optimum alignment between two sequences through the computation of local distances between the points in the temporal sequences. The algorithm proceeds by iteratively filling in values for each cell (i,j) in the $|X|$ by $|Y|$ cost matrix D. The value of the cell (i,j) is given by $D(x_i, y_j)$ which corresponds to a cost metric $Dist(i,j)$ to compute the distance of optimum warp path from (1,1) to (i,j) :

$$D(i,j) = Dist(i,j) + min(D(i-1,j), D(i-1,j-1), D(i,j-1))$$

computed using a cost metric $Dist(i,j)$ The figure below illustrates the working of the DTW algorithm.

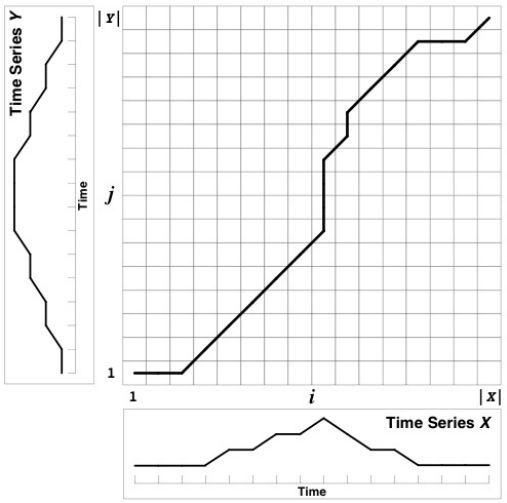


Figure 3.1: The DTW cost matrix with the minimum-distance warp path traced through it.

Generally the Euclidean metric is widely used as a cost metric. For comparing individual points between time series sequences, the metric is defined as: $Dist(i,j) = (x_i - y_j)^2$. For this project, I will be thus considering the DTW algorithm equipped with the euclidean metric as my baseline **DTW algorithm**. An outline of the base line DTW algorithm is given below:

Algorithm 1 Base line DTW

```
1: procedure BASE LINE DTW(seq1, seq2)                    ▷ two raw sequences
2:     DTW= zeros(length(seq1)+1,length(seq2)+1)
3:     for i=1: to length(seq1) do                  ▷ Initialise the DTW cost matrix
4:         DTW(i,0) = ∞
5:     end for
6:     for i=1 to length(seq2) do
7:         DTW(0,i) = ∞
8:     end for
9:     for i=2 to length(seq1) do
10:        for j=2 to length(seq2) do                ▷ cost(a,b)≡euclid(a,b)
11:            DTW(i,j) = cost(seq1(i),seq2(j)) + min{ DTW(i-1,j)+DTW(i,j-
   1)+DTW(i-1,j-1)}
12:        end for
13:    end for
14:    return result = DTW(n,m)/nm                  ▷ n=length(seq1), m=length(seq2)
15: end procedure
```

The computational complexity of the DTW algorithm is $O(n^2)$ where n denotes the length of the sequences that are being compared. Thus for time series domains having high dimensions i.e. long lengths, the time and computational costs incurred by the algorithm are quite high. To address this issue, two well-known global window constraints are employed: the Sakoe-Chiba band [7] and the Itakura parallelogram [12] . Figure 3.2 gives an illustration of the use of both constraints:

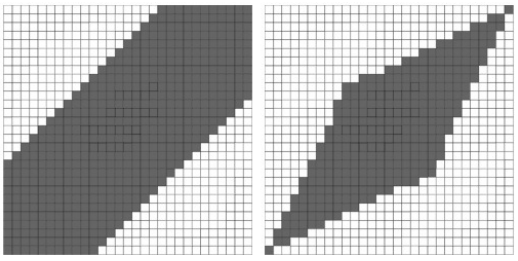


Figure 3.2: Two constraints: Sakoe-Chuba Band (left) and an Itakura Parallelogram (right), both have a width of 5.

The shaded areas in the above figure represent the cells of the cost matrix that are filled in by the DTW algorithm for each constraint. The width of each shaded area/window, is specified by a window parameter. When these

constraints are used, the DTW algorithm finds the optimal warp path only through the constraint window. However, the globally optimal warp path will not be found if it is not entirely inside the window[6]. The window parameter therefore determines the accuracy of the DTW algorithm. Decreasing the width decreases the accuracy. If the width is 0 and the series are of equal length then the DTW generalises to the Euclidean distance which is a notoriously inaccurate similarity measure for time series data[17]. To constraint the time complexity to a minimum, the vast majority of the data mining researchers use a Sakoe-Chiba Band with a 10% width. This reduces the time complexity from O(n^2) to O (wn) where w is the size of the window parameter. For finding motifs in datasets comprising of speech utterances, the use of such rigid window constraints is not always desirable. As a result of the same lexical identities spoken in difference contexts, by different speakers, in different environments etc, the optimal alignment paths may not always lie within the regions bounded by the widow constraints[4]. Thus it is necessary to explore alternative techniques to window constraints that can reduce the time complexity of DTW without degrading its accuracy. In the following chapter, I explore domain-dependent preprocessing methods to improve the performance of the DTW algorithm.

Chapter 4

Domain-dependent methods for improving DTW

Before investigating methods to improve the DTW algorithm itself, it is highly necessary to first understand the nature of the data sequences that the DTW is presented with. Achieving a thorough understanding of the data can result in the extraction of a smaller set of relevant features that can be used to achieve better discrimination between different classes/motifs. In this chapter, I investigate domain-dependent preprocessing techniques to improve the performance of the baseline DTW algorithms in clustering patterns that have long lengths.

There are presently two groups of preprocessing techniques commonly used to address this issue:

- Feature Selection
- Feature Extraction

Feature selection techniques involve selecting only a subset of attributes from the original data. With respect to the time series data, the technique is analogous to sub-sampling the sequence. To remove redundant features, one of the most popular feature selection approach is the exploratory data analysis(EDA). EDA is an approach to data analysis that postpones the usual assumptions about what kind of model the data follows with the more direct approach of allowing the data itself to reveal its underlying structure and models. The particular techniques employed in EDA are often quite simple, consisting of various techniques of:

1. Plotting the raw data such as data traces, histograms, histograms, probability plots, lag plots, block plots, and Youden plots.
2. Plotting simple statistics such as mean plots, standard deviation plots, box plots, and main effects plots of the raw data.
3. Positioning such plots so as to maximise our natural pattern-recognition abilities, such as using multiple plots per page.

Apart from removing redundant features, we can also construct more useful features from the existing ones. Feature extraction processes are concerned with the range of techniques that apply an appropriate functional mapping to the original attributes to extract new features. The intuition behind feature extraction is that the data vectors $\{x_n\}$ typically lie close to a non- linear manifold whose intrinsic dimensionality is smaller than that of the input space as a result of strong correlations between the input features. Hence by using appropriate functional mapping, we obtain a smaller set of features that capture the intrinsic correlation between the input features. By doing so, we move from working in high dimensional spaces to working in low dimensional spaces. The choice of appropriate functional mapping can also improve the clustering of data. For example, lets consider figure 4.1: The left- hand plot represents the locations of two dimensional data points in the original input space. The colours red and blue denote the classes to which the data points belong to. To cluster the data with respect to their classes, it will be ideal if we can partition the input space into disjoint regions where intra class variation is small and inter class separation in large. For this example, this is achieved by mapping the points to a feature space spanned by two gaussian basis functions(shown on the right). Now, we can partition the feature space into two disjoint regions, one of each cluster.

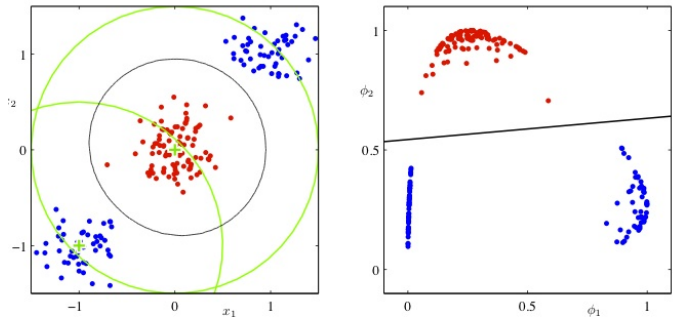


Figure 4.1: The figure on the right corresponds to location of the data points in the feature space spanned by gaussian basis functions $\phi_1(x)$ and $\phi_2(x)$

In the following sections, I investigate whether the application of exploratory data analysis and the construction of features that integrate metadata mainly knowledge of the domain can aid the baseline DTW algorithm is achieving low run times and good accuracy on long time series sequences. The primary data set that I will be using is the TIDIGITS corpus as the time series sequences of this dataset have lengths that are in the order of 10^4.

4.1 Feature Selection

The computational and time complexity associated with the DTW algorithm is governed by the length of the time series. Removing redundant features can result in a great reduction in the time complexity without any negative impact on the accuracy. To get an idea about the underlying structure of the data, I studied the plots of the time series sequences along with listening to the individual samples. Figure 4.2 shows the plot of raw signal corresponding to the word '8' by a speaker from the *boy* category.

From the visual and auditory analysis, I have made the following observations:

- Long durations of silence occupy the beginning and end of each utterance. The durations of the interesting regions that actually contain information about the spoken digit is quite small is comparison to the durations of silence regions. Removing these silence segments allow reduction in the dimensionality of the time series sequences with minimal loss of information.
- The recordings are highly distorted when played in *matlab*. The distorted signals fail to provide any type of auditory clue about category of the speaker i.e whether the speaker belongs to { boy,girl, men,women} the signal has to be played multiple time to correctly identify the spoken word.

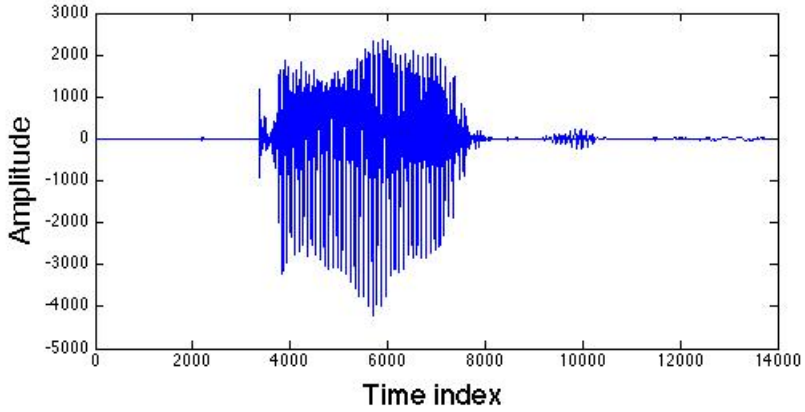


Figure 4.2: 'Raw 'signal

4.1.1 Signal Filter

To remove the redundant attributes from the time series utterances, I have constructed the following algorithm: 'SignalFilter' that performs feature selection by removing segments corresponding to durations of silence. An outline of the algorithm is as follows:

Algorithm 2 SignalFilter

```
1: procedure SIGNALFILTER(signal)                          ▷ raw signal
2:     threshold = 0
3:     maxAmplitude= max(rawSignal)
4:     Adapt the threshold based on the value taken by the maximum amplitude
5:     output← removeSilence(rawSignal,threshold)
6:     return output
7: end procedure
```

The algorithm employes an adaptive threshold to select and remove redundant attributes. The value of the threshold is dependent on the maximum amplitude of the sequence. The value set is comparatively higher for utterances of speakers having a loud and deep voice and lower for utterances for speakers having gentle and low voice.

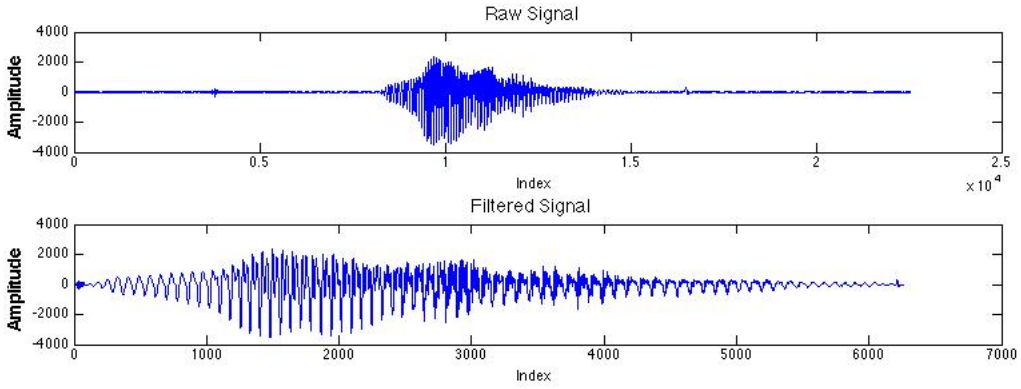


Figure 4.3: shows the raw acoustic signal corresponding to the utterances of the digit '8' alongside with the filtered signal at the bottom.

From figure 4.3, it can be observed that the filter preserves the interesting patterns associated with the utterance while succeeding in reducing the dimensionality of the data. To investigate the effect of introducing this prior feature selection step on the performance of the baseline DTW algorithm, I conducted the following experiment:

SETUP:

- Dataset used: TIDIGITS Test-set size : 976 samples

category	sample size
boy	162
girl	162
men	326
women	326

  Training data set size: 1467

category	sample size
boy	225
girl	234
men	495
women	513

- **Objective**: Investigating the effect of removing silence regions from the raw utterances on the accuracy and run time of the 1NN classifier using the base line DTW algorithm.
- **Algorithm:**

The main objective of my research is to improve the accuracy of the DTW algorithm while reducing the time and computational cost to a **minimum**. Even after applying the feature selection process, from initial experiments on few samples, I have found that the computational cost incurred by the algorithm is still very high. Hence to minimise the run time, I employed the Sakoe-Chuba band that has an adaptive window size of : w = max($\lceil 0.1 * max(n.m) \rceil$, abs(n-m)).

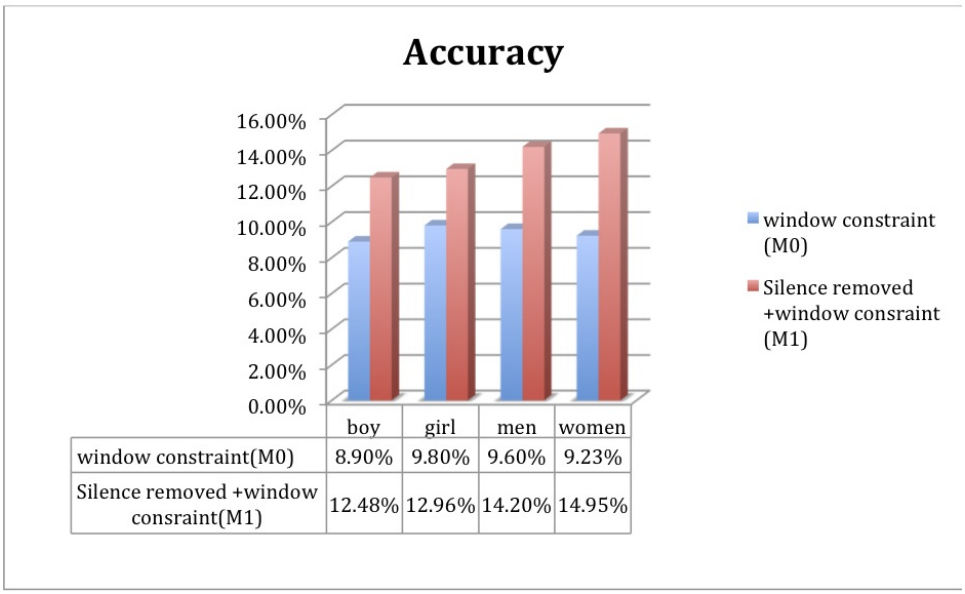
The lower bound of the window size is set to 10% of the size of the longest sequence which is the standard width used by the vast majority of the data mining researchers [20] to keep the time complexity of DTW to a minimum.

An outline of DTW algorithm augmented with the adaptive window constraint is given below:

Algorithm 3 Value-Based DTW

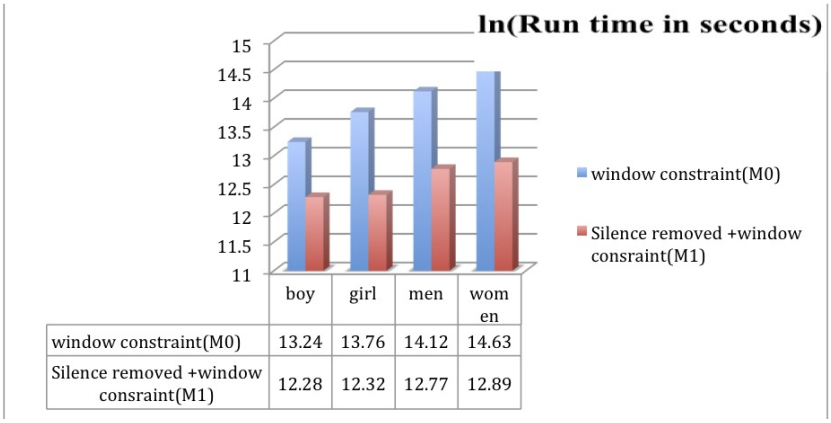
1: **procedure** VALUE-BASED($seq1, seq2$) ▷ two raw sequences
2: DTW= zeros(length(seq1)+1,length(seq2)+1)
3: w = max($\lceil 0.1 * max(n.m) \rceil$, abs(n-m)) ▷ Window constraint
4: **for** i=1: to length(seq1) **do** ▷ Initialise the DTW cost matrix
5: DTW(i,0) = ∞
6: **end for**
7: **for** i=1 to length(seq2) **do**
8: DTW(0,i) = ∞
9: **end for**
10: **for** i=2 to length(seq1) **do**
11: **for** j=max(2, i-w) to min(length(seq2), i+w) **do** ▷ cost(a,b)≡euclid(a,b)
12: DTW(i,j) = cost(seq1(i),seq2(j)) + min{ DTW(i-1,j)+DTW(i,j-1)+DTW(i-1,j-1)}
13: **end for**
14: **end for**
15: **return** result = $\frac{\text{DTW(n,m)}}{nm}$ ▷ n=length(seq1), m=length(seq2)
16: **end procedure**

- RESULTS:



	boy	girl	men	women
window constraint(M0)	8.90%	9.80%	9.60%	9.23%
Silence removed +window consraint(M1)	12.48%	12.96%	14.20%	14.95%

Figure 4.4



	boy	girl	men	wom en
window constraint(M0)	13.24	13.76	14.12	14.63
Silence removed +window consraint(M1)	12.28	12.32	12.77	12.89

Figure 4.5

Observations:

- The DTW algorithm achieves very poor accuracy. The poor accuracy is a result of one or a combination of the following three factors:
  - the use of raw values a features: the numerical value associated with each time point is not a complete picture of the data point in relation to the rest of the sequence.
  - the use of the rigid window constraint- the low accuracy may be attributed to the optimum warping path between similar patterns lying outside boundaries of the Sakoe-Chuba bands[5, 4].

– not integrating knowledge of the domain: The data set is comprised of speech utterances. It is a widely known fact that for speech data, the MFCC feature vectors capture the information of phones that make up a word. Since different lexical identities are composed of different phones, these use of MFCC vectors can increase the inter class variance of the samples. (details of MFCC to follow)[4, 21, 5, 22, 23].

- Removing 'silence' segments improve **both** the accuracy and the run-time of the algorithm. While the reason for the decrease in run-time is quite obvious, the increase in accuracy is not. All utterances contain durations of silence. Including the silence regions decreases the inter class variance and in turn increase intra class variance as they bring in an unwanted notion of similarity in dissimilar patterns.

4.1.2 Downsampling

From conducting further exploratory data analysis, I have observed that if I down-sample the utterances by $\frac{1}{2}$ which in other words corresponds to decreasing the sampling frequency by half, the global shape is still preserved even though some local information is lost as shown in figure 4.6. This results in further reduction in the dimensionality of the sequences i.e the length of the sequences.

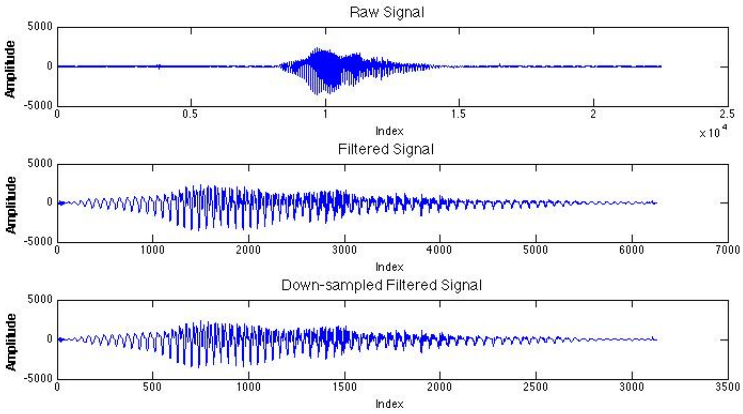
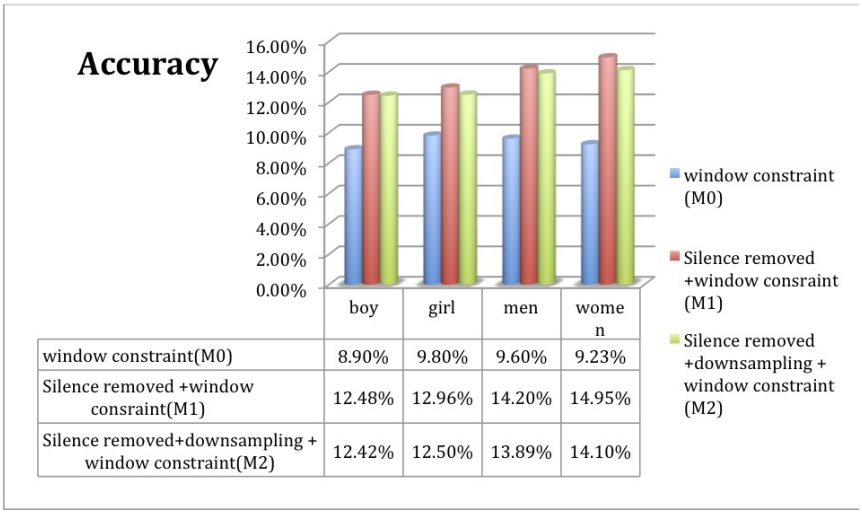


Figure 4.6: shows the raw acoustic signal of the digit '8' (top figure), the silence removed version of the signal(middle) and the silence removed and down sampled version of the acoustic signal (bottom)

To investigate the effect of performing further dimensionality reduction on the time series sequences through down sampling on the accuracy of the DTW, I performed the following experiment:

Dataset: The TIDIGITS training and set used in 4.1.1
Algorithm : baseline DTW augmented with an adaptive window constraint(4.1.1)
Approaches being compared: Removing silence utterances which we denote as method **M1** VS Removing silence utterance + downsampling VS baseline which we denote as method **M2**.

Results:

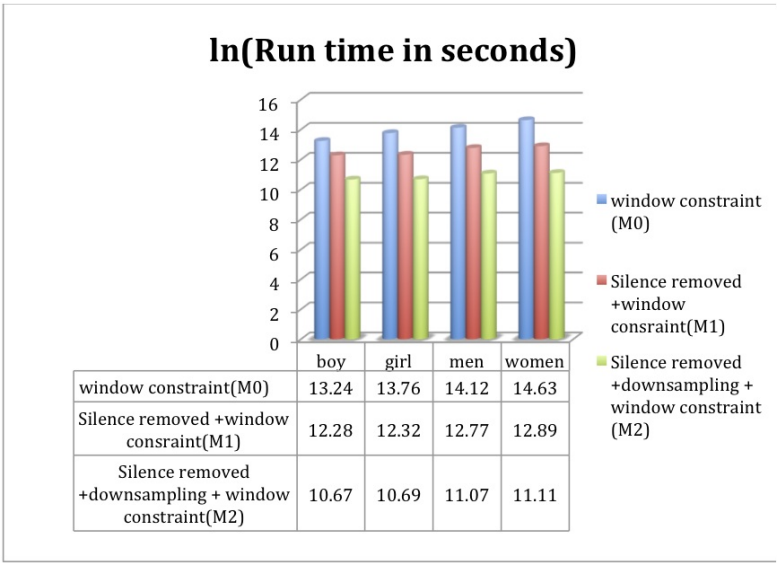


	boy	girl	men	women
window constraint(M0)	8.90%	9.80%	9.60%	9.23%
Silence removed +window consraint(M1)	12.48%	12.96%	14.20%	14.95%
Silence removed+downsampling + window constraint(M2)	12.42%	12.50%	13.89%	14.10%

Figure 4.7

Observations:

Performing the two stage feature selection step that involves removing 'silence' utterances followed by downsampling allow DTW to still construct more optimal paths that are aligned along the main diagonal for similar patterns than using the entire 'raw' time series sequences. This procedure results in a 4% increase in accuracy on average. Furthermore, the loss of local information through downsampling the non-silence regions leads to a minimal decrease in the accuracy of the algorithm in comparison to using the entire non silence regions for pattern matching. This supports the claim that the classes are differentiated mainly by their global shape.



	boy	girl	men	women
window constraint(M0)	13.24	13.76	14.12	14.63
Silence removed +window consraint(M1)	12.28	12.32	12.77	12.89
Silence removed +downsampling + window constraint(M2)	10.67	10.69	11.07	11.11

Figure 4.8: Integrating downsampling decreases the ln of the run time that is measured in seconds

Augmenting the down sampling procedure to the feature selection process leads to a further decrease of 1.5 in the ln(run time) of the DTW algorithm in comparison to just removing silence regions as a preprocessing step. This is expected since in the first stage of the preprocessing phase, redundant features are dropped which reduces the dimensionality(i.e length) of the sequences. The length of the sequences is reduced even further through downsampling the the output sequences of stage 1. Since the computational cost of DTW is directly dependent on the length of the sequences, thus decrements in dimensionality leads to a decrease in the computational cost.

4.2 Domain Dependent Feature extraction

From the analysis conducted so far, it can be concluded that heuristically selecting only significant attributes from the time series sequences does **improve** both the accuracy and the speed of the DTW. However, from the observation of the experimental results gathered so far, it is quite clear that the accuracy of the algorithm is very low. So far, I have investigated the effect on using 'raw' values of the time series sequences on the performance of the DTW. In this section, I investigate the contribution of using the window constraint and the contribution of using domain dependent features on the performance of the DTW algorithm. There are two motivations behind conducting this analysis:

- The primary motivation is to improve the speed of the DTW algorithm without degrading the accuracy. Choosing an appropriate functional mapping, can help map the data to a lower dimensional feature space that can capture the intrinsic qualities of the data. Thus constructing appropriate functional mappings can achieve both dimensionality reduction and higher accuracy.
- The other motivation is to investigate to what degree is the low accuracy error credited to not using domain dependent features and to using a rigid window constraint has on the low accuracy(The features that we have considered so far are the raw values indexed by time). Understanding the underlying factors that that influence the performance of the algorithm can provide insights on what aspect of the algorithm needs to be changed to gain better performance.

4.2.1 Mel Cepstrum Cepstral Coefficients

The primary dataset that I am working for this part of my project is the TIDIGITS corpus which is composed of speech utterances. For speech, the most commonly used features are the MFCC features-mel cepstrum cepstral coefficients. This feature representation is based on the idea of the cepstrum. For human speech, a speech waveform is created when a glottal source waveform of a particular frequency is passed through the vocal tract which because of its shape has a particular filtering characteristic. The exact position of the vocal tract is in fact the key attribute in providing useful information about phones(units of sounds). Cepstrum provides a useful way to separate the information of the vocal tract from the glottal source.

An outline of the MFCC feature extraction is given below:

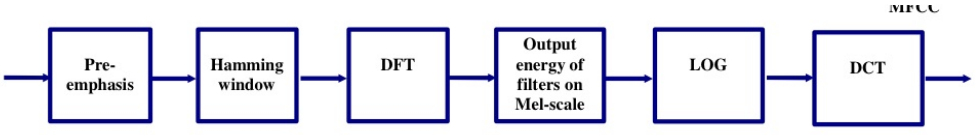


Figure 4.9: MFCC feature extraction

i Pre-emphasis: boosts the energy of the signal at high frequencies to improve phone detection

ii Windowing: partitions the time series sequence into frames using a hamming window

iii DFT: extracts spectral information at different frequency bands

iv Mel scale : transforming to mel scale improves recognition performance by allowing the model to take into account the property of human hearing

v Log : makes the feature less sensitive to variations in input such as power variations on the speakers mouth.

vi Cepstrum : separate the information of the vocal tract from the glottal source. The first 12 cepstral values from spectrum of the log of the spectrum are used

The windowing process reduces the length of the time-series sequences. Each sequence is segmented into frames of length 20 to 30 ms. The individual frames are then mapped using the procedure above to MFCC feature vectors where each MFCC vector is extracted from a particular frame. These feature vectors capture information about the positions of the vocal tract of speakers which as I have stated above is a key attribute in identifying phones. Since the result sequence of vectors is much smaller than the length of the original sequence, the size of the DTW cost matrix is much smaller than before which in turn lowers the time and computation cost incurred by the algorithm. The time complexity of DTW is now reduced from O(n^2) to O (u^2) where $u = \frac{n}{l}$ (l denotes the length of the frame.

The experiments conducted in section 4.1.1 and 4.1.2, have shown that the DTW algorithm performs very poorly in terms of accuracy on the TIDIGITS test data when it employs the narrow Sakoe-Chuba band to minimise the time complexity. The reason for this low accuracy was credited to the influence of one or a combination of the following factors : using a narrow window constraint, using raw attribute values and not using domain-dependent features. Having already investigated the influence of using raw values(4.1), for this part of the project, I constructed the following 3 methods in order to investigate the relative isolated impacts of the other two factors. The methods were tested on the TIDIGITS dataset, thus allowing the results to be compared with the results of the previous experiments so far :

i The first method denoted as **M3** employs the MFCC feature extraction mapping as a preprocessing step. The extracted features are used by

the valued-based DTW algorithm augmented with the adaptive Sakoe-Chuba band constraint described in 4.1.1 to cluster similar patterns. The performance of this method can be compared to the performance of the base line method **M0** to measure the relative impact on the accuracy and run-time of the DTW of replacing each numerical value associated with each time index with a domain dependent feature vector.

ii The second method denoted as **M4** employs a two stage preprocessing step. The feature selection procedure discussed in 4.1.1 is first applied to remove redundant features followed by MFCC feature extraction that achieves further dimensionality reduction. In this methodology, dimensionality reduction occurs at both stages of the pre-processing step. The downsampling method discussed in 4.1.2 was deemed not necessary as a feature selection step since the feature extraction phase allows further reduction in dimensionality without any loss of information. The sequence of vectors are then used by the DTW algorithm to find optimal warping along the main diagonal of the DTW cost matrix.

   By comparing the results of this method with the results of **M3** and **M1**, the optimum preprocessing strategy can be determined.

iii The third method denoted as **M5** is identical to **M4** with the exception that this version does not employ the window constraint. The main purpose of using this method is to check the relative impact of using window constraints. The difference in the performance between using method **M4** and **M5** can be used to check to what degree is employing such a rigid window constraint affects the accuracy and run-time of the algorithm.

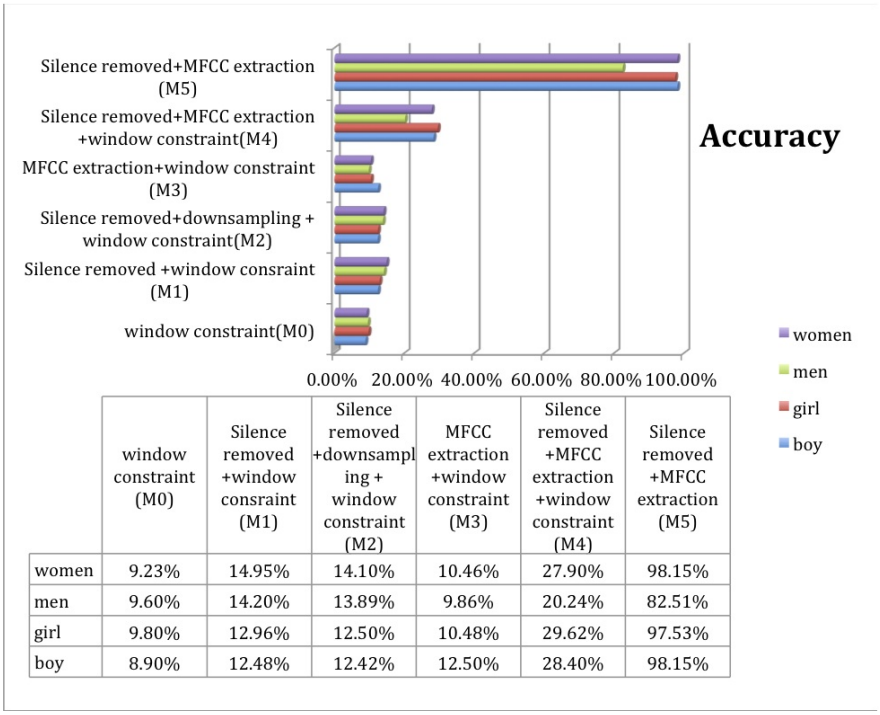
   The reason why I opted not to compare the performance of the classifier when it uses the baseline DTW algorithm(Chapter 3) with raw features against when it employs the window constrained DTW(discussed in 4.1.1) with raw features is that this particular dataset contains 'raw' sequences whose average length is about 7000. If no window constraint is applied, the run time incurred by the 1NN classifier using the baseline DTW algorithm on the 'raw' sequences will be very high. Thus to investigate the improvement in the accuracy of the DTW algorithm, without being disadvantaged with high run times, I have compared the accuracies of 1NN classifier when it employs method **M4** and when it employs method

M5 instead.

Experimental setup:

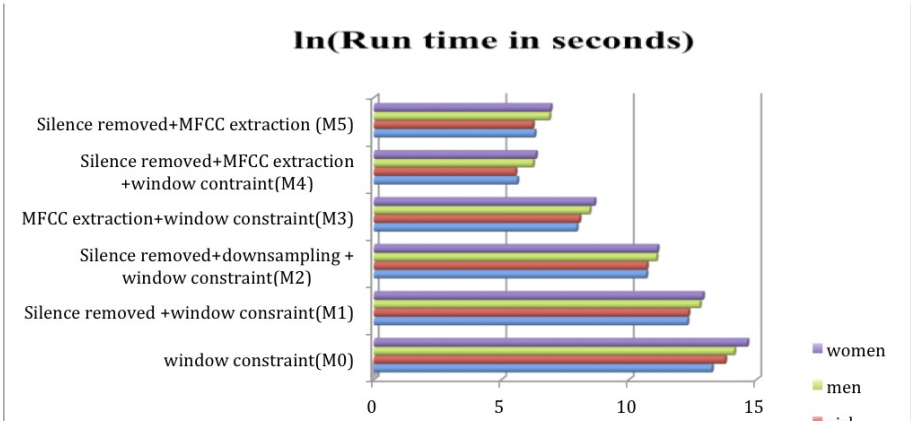
Dataset : The TIDIGITS training and test set (Chapter 2, 4.1.1)

RESULTS:



	window constraint (M0)	Silence removed +window consraint (M1)	Silence removed +downsampling + window constraint (M2)	MFCC extraction +window constraint (M3)	Silence removed +MFCC extraction +window constraint (M4)	Silence removed +MFCC extraction (M5)
women	9.23%	14.95%	14.10%	10.46%	27.90%	98.15%
men	9.60%	14.20%	13.89%	9.86%	20.24%	82.51%
girl	9.80%	12.96%	12.50%	10.48%	29.62%	97.53%
boy	8.90%	12.48%	12.42%	12.50%	28.40%	98.15%

Figure 4.10: Accuracy achieved by combining 1NN classifier using the DTW distance metrics with the various methods



	window constraint (M0)	Silence removed +window consraint (M1)	Silence removed +downsampling + window constraint (M2)	MFCC extraction +window constraint (M3)	Silence removed +MFCC extraction +window contraint (M4)	Silence removed +MFCC extraction (M5)
women	14.63	12.89	11.11	8.64	6.34	6.92
men	14.12	12.77	11.07	8.45	6.23	6.87
girl	13.76	12.32	10.69	8.05	5.54	6.22
boy	13.24	12.28	10.67	7.94	5.61	6.29

Figure 4.11: The ln of the run time(measured in seconds) incurred by the 1NN classiier

Observations:

- By comparing the baseline method **M0** with method **M3**, we can observe that making DTW domain-dependent has little significance on the accuracy when constraints are employed to minimise the time complexity.

  -In comparison to the base-line method **M0**, the accuracy of the window-constrained DTW only increases by 1.3% on average across all 4 categories.

  -The run-time on the other hand has improved considerably. The average ln(run time) has decreased by 5.7. This is obvious since the acoustic signal is mapped to smaller sequence of frames.

- Removing redundant features have greater relative impact on the accuracy than employing domain dependent MFCC features. This is evident if we compare the accuracy of the 1NN classifier when it uses the method **M2** against the accuracy achieved when it uses method **M3**. The presence of silence regions force the optimal warping paths between patterns of the same lexical identity to lie outside the regions in the cost matrix bounded by the Sakoe-Chuba band. This forces the DTW to find an

approximate optimal path with in the allowed regions which in turn leads to an increase in the error rate.

- Combining attribute/feature selection with MFCC feature extraction as a preprocessing step (i.e methods **M4** and **M5**) achieves greater improvement in accuracy and speed than using either of these approaches alone.

  -In comparison to just using MFCC feature extraction as a preprocessing step(method **M3**), the accuracy has been boosted up by 15.17% on average while the $ln(time)$ have been reduced by 2.36.

  – Similarly, in comparison to just using silence removal as a preprocessing step i.e method **M1**, the accuracy has increased by 12.9% on average while the $ln(time)$ have been reduced by 6.5.

- From the above observation alone, we can deduce two facts for this dataset, one of which is not that obvious: removing redundant features have a greater relative impact on improving the performance of a constrained DTW than employing domain dependent features and the second fact which is quite obvious is the strong dependency between the algorithm's accuracy and the size of the window constraint. It can be concluded that the use of the rigid window constraint is the main contributing factor for achieving small accuracy rates.

  Since the main goal is to improve **both** the accuracy and speed of DTW in handling sequences of high-dimensionality i.e long lengths, removal of silence segments provides an ideal mechanism to improve **both** the time complexity and the accuracy of the algorithm. Dropping the window constraint on the other hand, will although lead to considerable increase in the accuracy by will result longer run-times. This is not ideal when working on large high dimensional datasets.

- Employing Method **M5** achieves almost near perfect accuracy. Dropping the window constraint of the DTW improves the accuracy of the 1NN classifier by 67.15% from the accuracy achieved by the classifier when it employs method **M4**. In other words 67.15% of the time on average, the optimum warping path lie outside the regions bounded by the Sakoe-Chuba band constraints, This confirms that patterns belonging to the same lexical identity can have an *overly temporal askew* between them as result of the different contexts in which the words are spoken and/or as

a result of different speakers speaking the same word.

- The run time incurred by the 1NN classifier using method **M5** is lower than the run time of all proposed methods with the exception of using method **M4**. Method **M5** therefore provides the best balance in the fulfilling the two conflicting objectives of high accuracy and high speed that any of the other investigated methods.

Conclusion

To summarise, to improve the performance of the DTW algorithm:

- In 4.1, I have investigated ways to remove redundant features by conducting exploratory data analysis.
- In 4.2, I introduced domain dependent feature extraction methods and investigated the influence of the use of such features on the performance of DTW
- I have investigated the investigated the relative isolated impacts of using raw values, employing a rigid window constraint and using domain dependent features on the performance of the DTW algorithm
- I have observed that although the DTW algorithm is domain independent, augmenting the DTW with a domain dependent preprocessing technique can greatly improve its performance in terms of **both** accuracy and speed with out the need of any global constraints. For the TIDIGITS dataset, employing the preprocessing steps of 'silence' removal followed by MFCC feature extraction allows DTW to improve not only its accuracy but also speed even without the use of global window constraints.

Chapter 5

Domain Independent methods for Improving DTW

So far, we have investigated methodologies that improve both the speed and accuracy of the DTW algorithm by integrating meta data (i.e knowledge of the domain) in the preprocessing stage. The problem with such methodologies is that the same algorithm cannot be extended across multiple domains: the MFCC feature extraction discussed in the previous chapter is only applicable to time series sequences that correspond to speech utterances. To develop a framework that can be extended across multiple domains, it is necessary to use features that are entirely data driven. In the first half of this chapter, I investigate in detail a recently proposed data driven feature extraction methodology [9] that is aimed to improve the accuracy of the DTW across multiple domains. In the second half of this chapter, I investigate alternative measures to using window constraints that can improve the performance of the algorithm in terms of **both** time and accuracy across different time series domains.

5.1 Domain-independent feature extraction

Ideally, we require features that reflect information about the structure of the data. This allows the DTW to construct a complete 'picture' of the data point by capturing its relation to the rest of the sequence. Thus, by doing so we can achieve optimal alignments that are close to the diagonal for similar sequences. The fundamental problem of baseline (value-based) DTW is that the 'raw' numerical value associated with each point in the time series sequence is not

a complete picture of the data point in relation to the rest of the sequence. The context such as the position of the points in relation to their neighbours is ignored. To fix this issue, an alternative form of DTW known as *derivative* DTW[24] is proposed but it too fails to achieve better performance across all domains as it ignores to take into account the common sub-patterns between two sequences (mainly global trends).

To address these drawbacks, recent works have been conducted to construct good feature extraction methods that are entirely data-driven. One particular approach that has been seen to achieve good accuracies across multiple domains is the method proposed by Xie and Wiltgen in their paper [9]. The authors highlight a domain-independent feature extraction process where each point in the time series sequence is replaced by a 4 dimensional vector. This 4-d vector attempts to provide a complete picture of a point in the relation to the rest of the sequence. The first two features in this vector hold information of the local trends around a point while the last two features reflect information about the global trends. From experiments conducted on the UCR datasets, they have observed that embedding DTW with this feature extraction process yields greater accuracy across all datasets.

Definition of local feature given in [9] is as follows:

$$f_{\text{local}}(t_i) = (y_i - y_{i-1}, y_i - y_{i+1})$$

The first feature reflects the difference between the values of the current index and the previous index while the second feature corresponds to the difference between the value of the current index and the succeeding index.

The extraction of global features however, is constrained by two factors:

- the features must reflect information about global trends
- the features must be in the same scaling order as the local features. Being in the same scale allows them to be combined with local features.

In [9], the authors used the following method to extract global features from the time series sequence:

$$f_{\text{global}}(t_i) = (y_i - \sum_{k=1}^{i-1} \frac{y_k}{i-1}, y_i - \sum_{k=i+1}^{M} \frac{y_k}{M-i})$$

The first feature represents the deviation of the value at time i from the mean

of the values of the sequence that has been seen so far while the second feature represents the deviation of the value at time i from the mean of the values that is yet to be seen. This formulation allows the detection of significant 'drops' or 'rises' in the series.

Note : The local and global features have no definition for the first and last points in a sequence and to keep the terminology clear, when I refer to the dimension of the time series, I mean the length of the time series.

One of the drawbacks of using this feature extraction methodology is the absence of achieving dimensionality reduction. The length of the sequence of vectors is approximately the same as the dimensionality of the original sequence. The DTW algorithm combined with this feature extraction process therefore suffers from the curse of dimensionality as before. Since minimising the time complexity is a major priority for this analysis, the DTW algorithm is subjected to the adaptive Sakoe-Chuba band window constraint discussed in 4.1.1 to minimise the run time.

Xie and Witgen [9] have already shown that augmenting this feature extraction methodology to the DTW algorithm improves its accuracy across different domains. However, in their analysis due to the availability of sufficient computing power, they didn't use any window constraints when performing their experiments. For problem scenarios where availability of computing power is limited and the speed of the DTW is considered an equal priority along with the accuracy, it will be interesting to investigate whether this methodology can allow DTW to achieve better performance in terms of accuracy over the base line method.

To investigate the effect of introducing this prior feature selection step, I conducted the following 2 experiments:

Objective Compare the affect between using global and local features and using raw values on the performance of the 1 NN classifier employing a **window constrained** DTW algorithm.

- Experiment 1

  Datasets: TIDIGITS dataset(4.1.1)

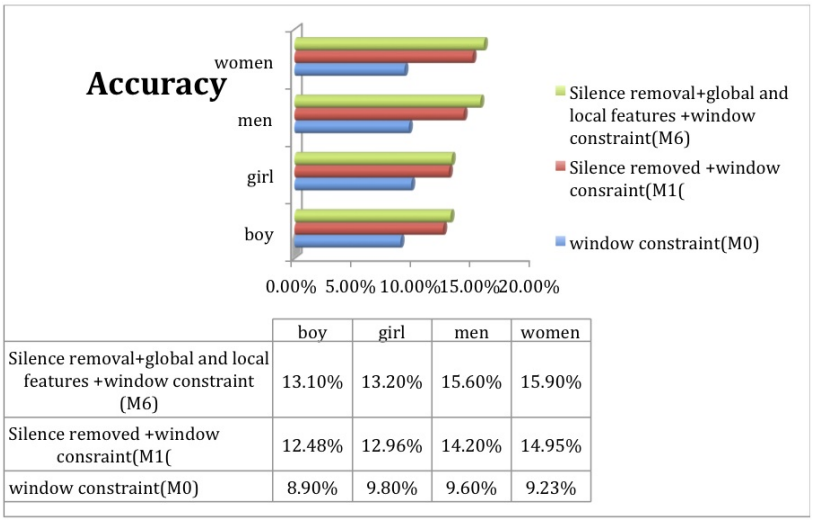
  Setup Prior to extracting these domain independent features, I performed the feature selection step (**M1**) which involves removing segments of utterances that correspond to silence. When conducting experiments

using MFCC feature vectors, removing redundancy allows the optimal warping paths between similar patterns to lie closer to the main diagonal. Using this feature selection step also has the advantage of dimensionality reduction.

By comparing the experimental results of this proposed method, which I denote as method **M6** for simplicity purposes, with the baseline method **M0** and method **M1**, we can observe whether for the TIDIGITS dataset, removing redundant features is more significant to using features that capture information about the global and local trends. Plus, this will allows us to investigate whether using such features allow a constrained DTW algorithm to achieve higher accuracies than using raw values.

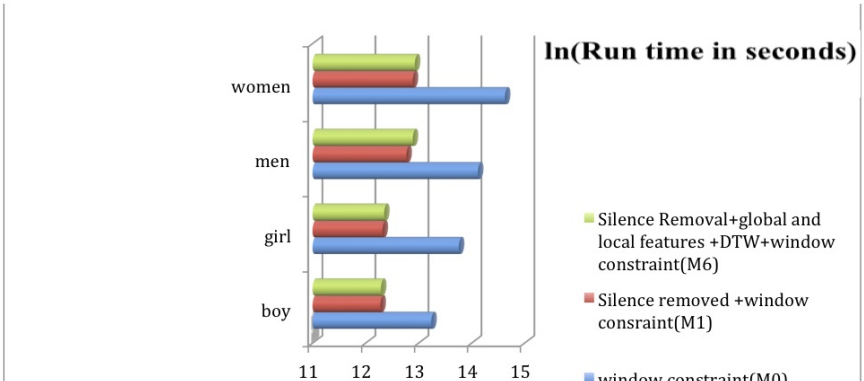
RESULTS

A summary of the results are given below:



	boy	girl	men	women
Silence removal+global and local features +window constraint (M6)	13.10%	13.20%	15.60%	15.90%
Silence removed +window consraint(M1(	12.48%	12.96%	14.20%	14.95%
window constraint(M0)	8.90%	9.80%	9.60%	9.23%

Figure 5.1



	boy	girl	men	women
Silence Removal+global and local features +DTW+window constraint(M6)	12.29	12.35	12.89	12.93
Silence removed +window consraint(M1)	12.28	12.32	12.77	12.89
window constraint(M0)	13.24	13.76	14.12	14.63

Figure 5.2: ln of the run time(measured in seconds): $log_e(time)$

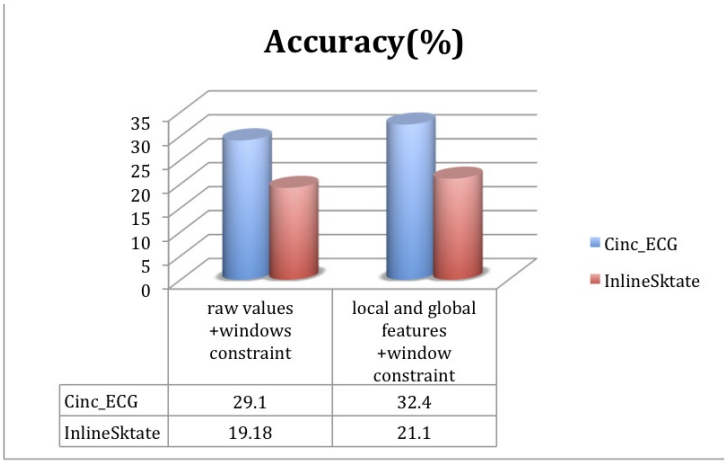
Observations:

- Removing redundant features i.e removing regions of silence is a greater contributing factor in improving the performance of a window constrained DTW than applying a feature extraction process that integrates meta data(**M3**) or that captures information about local and global trends(**M6**).
- The 1NN classifier combined with the DTW algorithm achieves very low accuracies for all 3 methods. As concluded from previous experiments, the low accuracy is primarily credited to the use of the rigid window constraint.
- The computational cost incurred by the 1NN classifier using method **M6** is higher than when it uses method **M1**. One possible explanation is that the cost of applying the euclidean metric on vectors >cost of applying the euclidean metric on points. Since the euclidean metric is applied mn times where m and n represent the length of the respective sequences, the overall computational cost therefore increases.

- Experiment 2

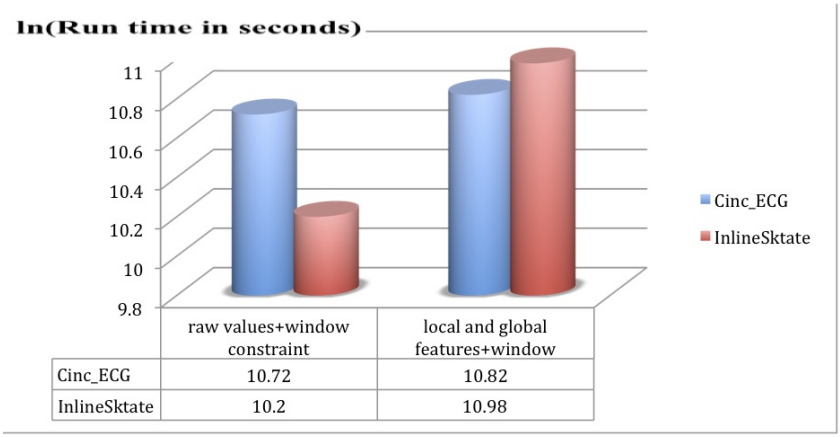
  Datasets: UCR datasets: InlineSkate and Cinc_ECG_Torso

The results are as follows:



	raw values +windows constraint	local and global features +window constraint
Cinc_ECG	29.1	32.4
InlineSktate	19.18	21.1

Figure 5.3: Using features that reflect information of trends improves the accuracy of the algorithm



	raw values+window constraint	local and global features+window
Cinc_ECG	10.72	10.82
InlineSktate	10.2	10.98

Figure 5.4: ln of the run time(measured in seconds): $log_e(time)$

Observations

- The differences between performances of the two versions of DTW are consistent with the observations made in the previous experiment.
- In comparison to the base line method, there is only a minimum improvement in the accuracy of the window constrained DTW. For the Cinc_ECG_Torso time series dataset, using global and local features improves the accuracy by only 3.2% whereas for the InlineSkate dataset, the accuracy improves by only 1.92%.

Conclusion:

The experiments conducted by Xie and Witgen[9] on the UCR datasets have shown that using their feature extraction methodology allow DTW to achieve much higher accuracies than using just 'raw' values . Optimal paths that closer to the diagonal region of the cost matrix have smaller distances than paths that are further away from the diagonal. The improved accuracy is credited to the construction of optimal paths that are closer to the diagonal than before for sequences belonging to the same class. However the results from their experiments do not provide any information on the degree of translation undergone by these warping paths. The results from my experiments provides this **insight**. Using the domain independent methodology only leads to a fractional increase in the accuracy of the DTW when it employs a window constraint. This shows that for similar patterns, the degree of translation undergone by the optimal warping paths is very small since very few ideal optimal paths enter the region defined by Sakoe-Chuba band.

The low accuracies of the DTW can be improved by increasing the width of the Sakoe-Chuba band. But increasing the width also leads to a reduction in speed. This provides the motivation to investigate alternative methods that can a better balance between the two conflicting goals of accuracy and speed. In the next section, I investigate a self-proposed method that is an alternative approach to using window constraints to speed up DTW. The new proposed methodology uses the data-driven feature extraction process that is discussed in the current section. The aim here to construct a domain independent framework that allows DTW to attain improvements in both speed and accuracy over the Sakoe-Chuba band window constrained DTW.

5.2 Extending DTW

The feature extraction methodology discussed in the previous section offers no advantage of dimensionality reduction. The time series sequence is mapped to sequence of vectors whose length is $\|X_n\| - 2$. (where $\|X_n\|$ denotes the length of the original time series sequence). The MFCC feature extraction process (4.2) on the other hand, does provide the advantage of dimensionality reduction. The time series sequence is first segmented into series of frames of length 20ms i.e 200 points. Through appropriate functional mapping, each frame is then mapped to a vector. Hence the time complexity of the DTW algorithm have been reduced from O(n^2) to O (u^2) where $u = \frac{n}{l}$ (l denotes the length of the

frame).

Using the MFCC feature extraction method as motivation, I propose the following methodology:

i Apply the feature extraction process of 5.1 to embed the information of local and global trends.

ii Partition the sequence of vectors into frames of width m. The default value of m is set to 50. Results from experiment 3 discussed later in the chapter have shown that choosing the default value to be 50 is a safe option in improving the DTW algorithm's accuracy and speed. For speech utterances in the TIDIGITS dataset, this is equivalent to having frames of size 5ms. Unlike the windowing process of MFCC extraction, the frames here correspond to sequence of vectors rather sequence of numerical values. This partitioning process reduces the length of the sequences by m times and hence decreases the time complexity of the DTW algorithm from $O(n^2)$ to $O(\frac{n}{m})^2$.

iii Adapt the cost metric of the DTW algorithm to work on series of frames rather than series of vectors

The problem can now be shifted to finding an appropriate kernel that can be used to compute the similarity between frames composed of feature vectors. Ideally, we want a metric that takes into account the variation of speed and time when comparing two similar subsequences. To understand why we need the metric to be invariant to variations in speed and time, lets consider the following two signals:

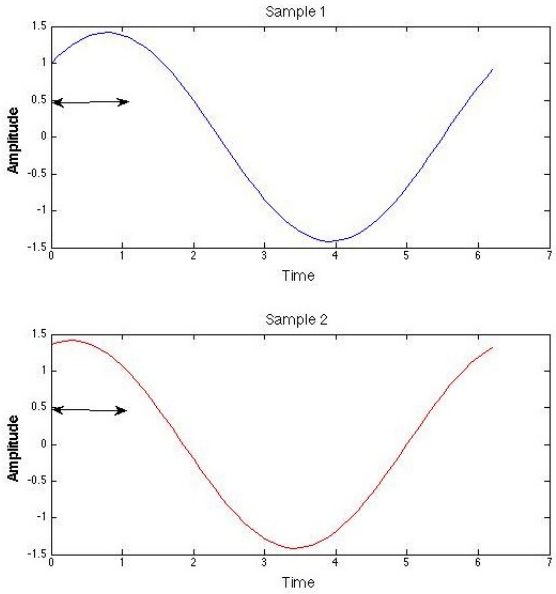


Figure 5.5: Two signals separated by translation

The signal denoted by the 'red' color is a 'slower' version of the signal denoted by the 'blue' color. Consider the frames corresponding to the subsequences annotated by the double arrows(see figure 5.5), in each of the two signals, in this scenario using the Euclidean metric to perform a linear comparison of the two frames is inappropriate. The Euclidean metric in this context is identical to linear time warping where the two subsequences/frames will be compared based on a linear match of the temporal dimensions of the two sequences/frames. In our context, we need a kernel that computes the similarity between two sub-sequences by warping the time axis. Intuitively speaking, the kernel must behave like a DTW algorithm that compares the global and local properties associated with a point in one frame with the global and local properties of all points in the second frame as illustrated by the figure below.

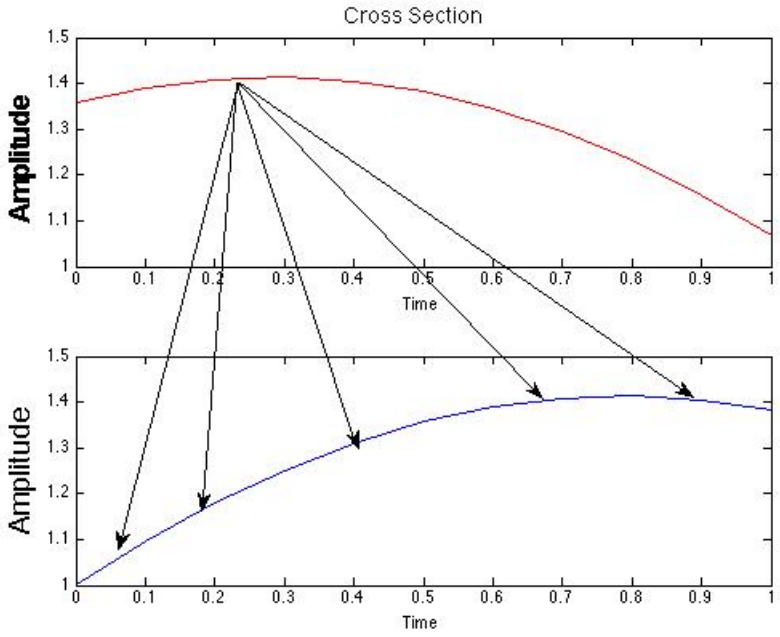


Figure 5.6: Two identical subsequences varying in time

The motivation behind my proposed kernel function comes from the polynomial kernel. Let x and z be two two-dimensional vectors and consider the simple polynomial kernel of degree 2 :$k(x,z) = (x^T z)^2$. This kernel can expressed as :

$$\begin{aligned} k(x,z) &= (x^T x')^2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\ &= (x_1^2, 2x_1 x_2, x_2^2)(z_1^2, 2z_1 z_2, z_2^2)^T \\ &= \phi(x)^T \phi(z) \end{aligned}$$

The 2nd order polynomial kernel is equivalent to a corresponding feature mapping $\phi(x)$ that maps a two dimensional vector to $(x_1^2, 2x_1 x_2, x_2^2)$ where each attribute is monomial of order 2 . Generalising this notion to order M then $k(x,z) = (x^T z)^M$ represents the sum of all monomials of order M. If we imagine x and z to be two images, then the polynomial kernel represents a particular **weighted** sum of all possible products of M pixels in the first image with M pixels in the second image.

Using this as motivation I propose the following kernel:.

$$k(x,z) = < \sum_{i=1}^{n} x_i, \sum_{j=1}^{n} z_j >$$

Here n denotes the length of the frame while x_i and z_j correspond to the individual 4-dimensional feature vectors that make up the frame. The above kernel corresponds to the dot product between the individual sums of the two frames. Although it may not look obvious, this construction actually allows the comparison between all feature vectors in the first frame with all feature vectors in the second frame as shown below:

$$\begin{aligned} k(x,z) &= < \sum_{i=1}^{n} x_i, \sum_{j=1}^{n} z_i > \\ &= < (x_1 + x_2 + x_3 + ..), (z_1 + z_2 + z_3 + ..) > \\ &= < x_1, z_1 > + < x_1, z_2 > + < x_1, z_2 > ... + < x_2, z_1 > + < x_2, z_2 > + ... \end{aligned}$$

From above expression, we can see that the proposed kernel corresponds to a sum of all possible dot products of pairs belonging to the set $\{(x_i z_i) | x_i \in \text{seq1}, z_i \in \text{seq2}\}$. It is easy to check that this proposed kernel is in fact a valid kernel function:

- K(x,z)= K(z,x) $\Rightarrow$the function is symmetric.
- The kernel can be written as a scalar product in feature space: K(x,z) $=\phi(x)^T \phi(x)$ where the feature mapping corresponds to a finite summation of vectors $\phi(y) = \sum_{i=1}^{n} y_i$.

Augmenting the kernel to the DTW algorithm allows DTW to work on long time sequences without using a window constraint. To use this kernel as an appropriate cost function in the DTW algorithm, we need a functional mapping that:

1. constraints the codomain to be in the range from 0 to ∞.
2. ensures larger values given by the function signify great degree of dissimilarity and smaller values signify a high degree of similitude.

An inspection of the kernel function shows the function represents the sum of dot products of all vectors in one frame with all vectors in the second frame. An ideal cost function that make use of dot products is the *arc-cosine*. Using an

appropriate substitution, I have thus constructed the following cost metric:

$$\theta = \frac{< X, Z >}{|X||Z|}$$

where θ is the *arc-cosine* distance, $X = \sum_{i=1}^{n} x_i$ and $Z = \sum_{j=1}^{n} z_i$.

A formal outline of the proposed methodology is given by the following algorithm:

Algorithm 4 Proposed DTW

procedure VALUE-BASED($seq1, seq2$) ▷ two sequences of feature vectors
  seq_1←segment(seq1,n) ▷ Segment the sequences using a window of size n
  seq_2←segment(seq2,n)
  for i=1: to length(seq_1) **do** ▷ Initialise the DTW cost matrix
    DTW(i,0) = ∞
  end for
  for i=1 to length(seq_2) **do**
    DTW(0,i) = ∞
  end for
  for i=2 to length(seq_1) **do**
    for j=max(2, i-w) to min(length(seq_2), i+w) **do**
      DTW(i,j) = $\theta = \frac{<X,Z>}{|X||Z|}$+ min{ DTW(i-1,j)+DTW(i,j-1)+DTW(i-1,j-1)}
    end for ▷ $X = \sum_{i=1}^{n} x_i$ and $Z = \sum_{j=1}^{n} z_i$
  end for
  return result = $\frac{\text{DTW(n,m)}}{nm}$ ▷ n=length(seq1), m=length(seq2)
end procedure

5.2.1 Testing the methodology

The primary objective is to improve the performance of the DTW algorithm in problem domains where minimising time complexity shares equal priority with improving accuracy. To investigate whether the proposed changes does provide a better alternative to using the Sakoe-Chuba band window constraint, I have conducted the following experiments:

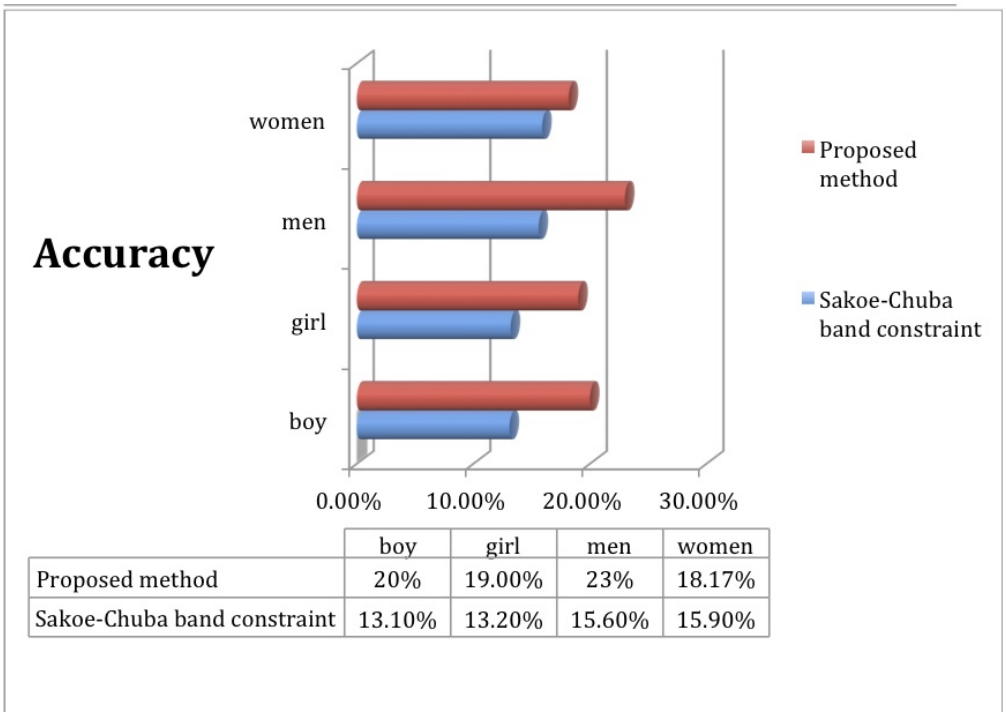
Experiment 1

Dataset used : The TIDIGITS dataset.

Method used : The preprocessing method discussed in 5.1 i.e **M6**: the framework consists of a two stage preprocessing step that involves feature selection process consisting of 'silence' removal followed by the domain independent feature extraction methodology proposed in [9]

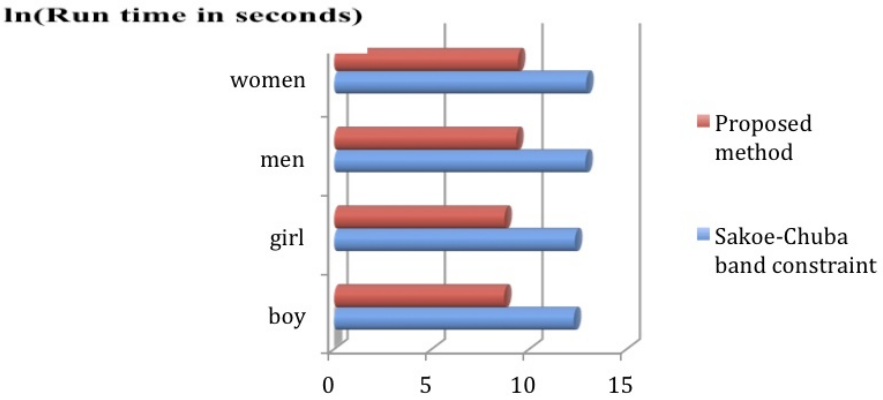
Variables being compared: Sakoe-Chuba band window constraint VS the proposed methodology

RESULTS:



	boy	girl	men	women
Proposed method	20%	19.00%	23%	18.17%
Sakoe-Chuba band constraint	13.10%	13.20%	15.60%	15.90%

Figure 5.7: Accuracy



	boy	girl	men	women
Proposed method	8.72	8.74	9.34	9.43
Sakoe-Chuba band constraint	12.29	12.35	12.89	12.93

Figure 5.8: ln of the run time(measured in seconds): $log_e(time)$

There are quite number of interesting observations that can made from figures 5.7 and 5.8.

- In comparison to employing the Sakoe-Chuba band window constraint, the alternative method allows DTW to improve its accuracy across all 4 categories by 6.54% on average.
- Equipped with the new changes, the run time of the algorithm has decreased exponentially by an **order** of 3.1 on average. The reduction of the time complexity is mainly due to the partitioning of the sequence into time slices of width 5 ms.The reduction in the length of the sequences by an order of 50 results in the shrinkage of the search space of DTW thus causing the algorithm to improves its speed.

The proposed methodology can also be adapted to use domain-dependent features. As I have stated earlier, the main motivation for constructing this framework is to provide better performance in terms of both accuracy and run-time for time series problem domains where the use of rigid window constraints is deemed necessary to minimise the computational cost. In the previous chapter, I have discussed that when the MFCC feature extraction is used as the only preprocessing step, the use of the adaptive window constraint was deemed necessary to minimise the run-time. However from the experimental results in the previous chapter, it was observed that enforcing such a

rigid constraint results in a high error rate. To verify that the new proposed changes to the DTW algorithm allow it to perform equally well when using domain dependent features, I have repeated the same experiment again but this time, as a preprocessing step I have just used the MFCC feature extraction i.e method **M3**.

A summary of the results are as follows:

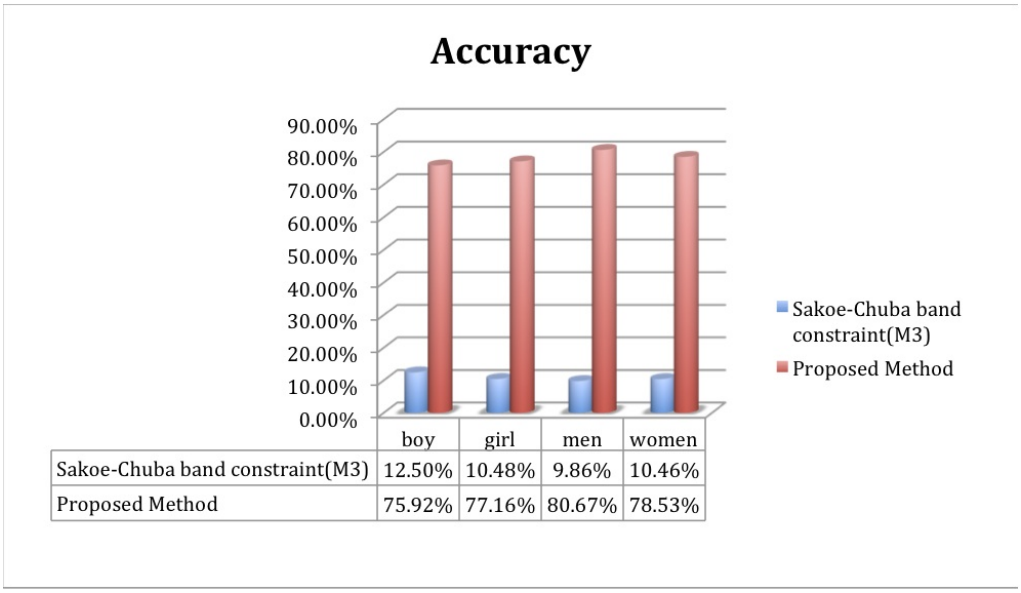
Experiment 2

Dataset used : The TIDIGITS test and training data.

Method : **M3** which consists of single preprocessing step that involves the extraction of MFCC features

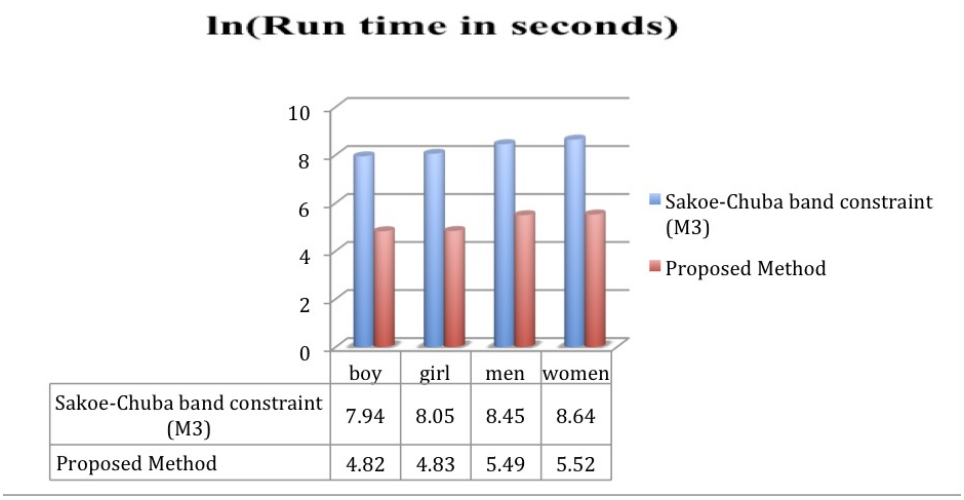
Variables being compared: employing window constraints vs the new proposed changes

Results



	boy	girl	men	women
Sakoe-Chuba band constraint(M3)	12.50%	10.48%	9.86%	10.46%
Proposed Method	75.92%	77.16%	80.67%	78.53%

Figure 5.9: Accuracy



	boy	girl	men	women
Sakoe-Chuba band constraint (M3)	7.94	8.05	8.45	8.64
Proposed Method	4.82	4.83	5.49	5.52

Figure 5.10: ln of the run time(measured in seconds): $log_e(time)$

Observation The results are consistent with the findings in the previous experiment. The accuracy of the DTW algorithm has improved by over 60% on average and the average run time has decreased exponentially by an order of 3.12. Hence in comparison to the window constrained DTW, the new framework takes better advantage of employing domain dependent features.

However, since the tests so far has been conducted on the TIDIGITS test set, it is possible that this new approach is only tailored for this particular dataset. Thus to confirm that the boost in the performance is not tailored for this particular time-series dataset, I have performed classification using the proposed DTW algorithm on the UCR datasets: InlineSkate and CINC_ECG_TORSO and compared the performance against the performance achieved by the 1 NN classifier using the baseline DTW metric augmented with Sakoe-Chuba band constraint discussed in 4.1.1.

Experiment 3

Datasets used : InlineSkate and CINC_ECG_TORSO

Setup :

Method used : The preprocessing method discussed in 5.1 i.e **M6**: the framework consists of a two stage preprocessing step that involves feature selection process consisting of 'silence' removal followed by the domain independent feature extraction methodology proposed in [9].

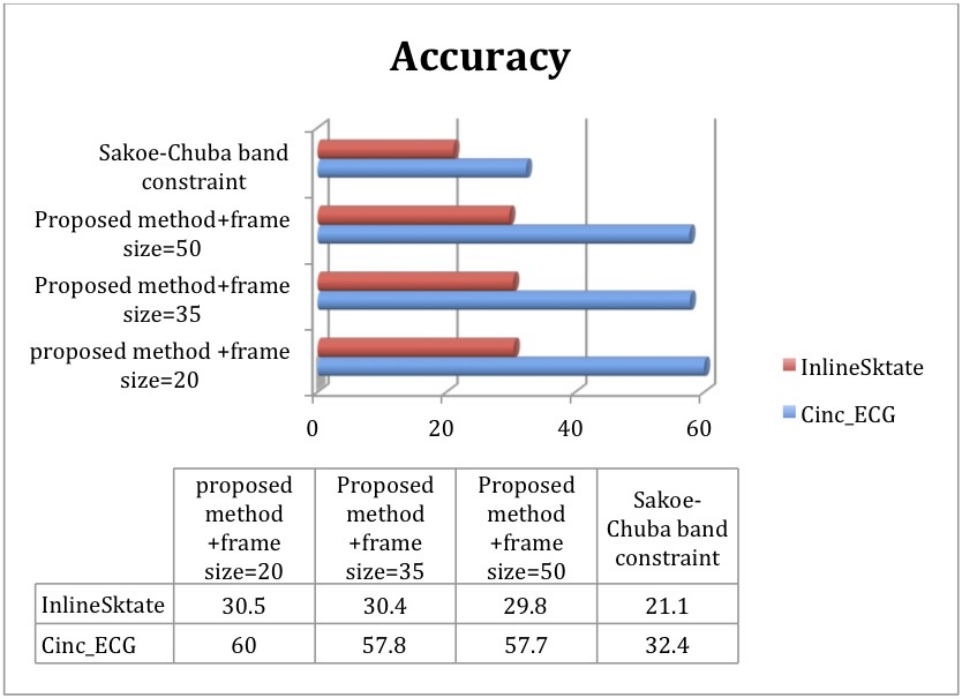
In the proposed approach, the width of the time slices has been kept fixed at a default value of 50 so far. In this experiment, I also investigate the influence of

this parameter on the performance of the DTW. Decreasing its value reduces the size of the time slices which in principal should increase both accuracy and time-complexity . The core kernel used by the new algorithm is based on the function:

$$k(x,z) = < \sum_{i=1}^{n} x_i, \sum_{j=1}^{n} z_j >$$

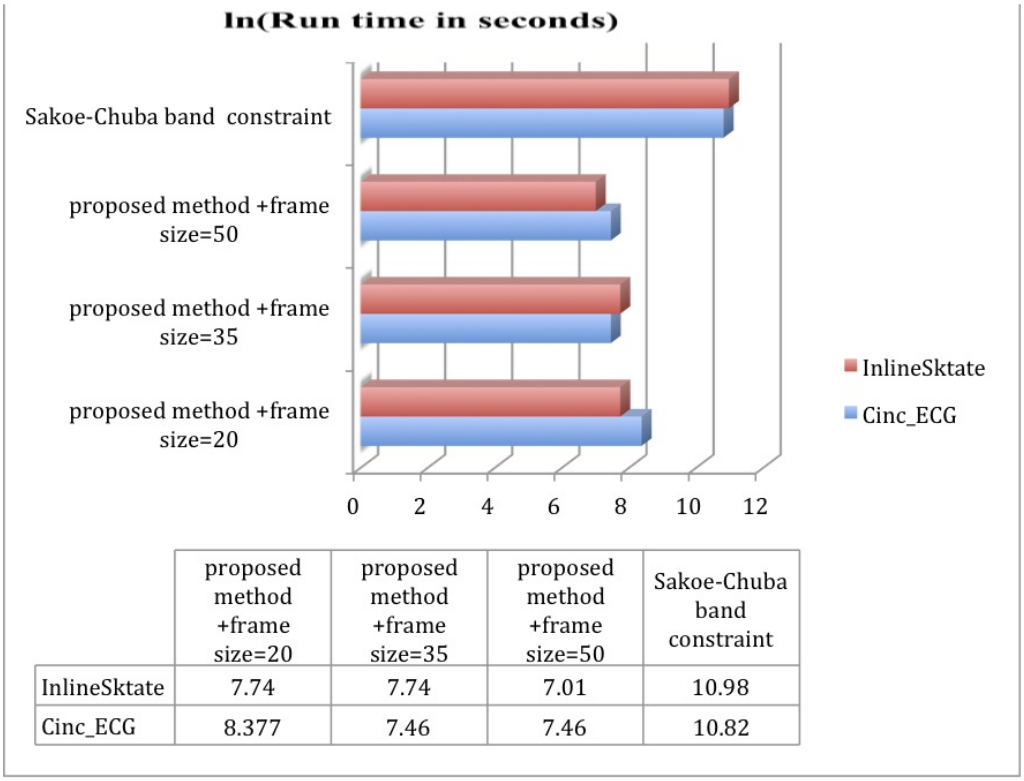
k(x,z) represents the sum of all possible dot-products. Using smaller subsequences allow the similarity measure to be dominated by the dot products of points whose local and global features are most alike. However, this suffers from the drawback of achieving lesser dimensionality reduction. Thus the time and computational complexity suffers.

RESULTS



	proposed method +frame size=20	Proposed method +frame size=35	Proposed method +frame size=50	Sakoe-Chuba band constraint
InlineSktate	30.5	30.4	29.8	21.1
Cinc_ECG	60	57.8	57.7	32.4

Figure 5.11: Accuracy



	proposed method +frame size=20	proposed method +frame size=35	proposed method +frame size=50	Sakoe-Chuba band constraint
InlineSktate	7.74	7.74	7.01	10.98
Cinc_ECG	8.377	7.46	7.46	10.82

Figure 5.12: ln of the run time(measured in seconds): $log_e(time)$

Observation

- The proposed changes do provide a **better** alternative to using rigid constraints for problem domains where the time complexity is of high priority and the use of window constraints can not be avoided. The accuracy of the DTW algorithm under the new methodology is significantly higher then employing a window constrained DTW on sequences of extracted local and global feature vectors.
- Decreasing the size of the time slices only leads to a minimal increase in accuracy. Thus using the default value of 50 seems to be safe option as the algorithm better accuracy and speed than using the rigid Sakoe-Chuba band constraint .

5.2.2 Conclusion

In chapter 3, we have seen that under the window constraint the time complexity of the algorithm decreases from O(n^2) to O(wn). From the experimental results, it can be observed that the new proposed changes allow DTW to

achieve run times that are exponential lower than the run times achieved by the window constrained DTW. Thus it can be concluded that the time complexity of the proposed DTW variant is smaller that O(wn).

We have seen so far that increasing the speed of DTW negatively impacts the accuracy of the algorithm . The new methodology actually provides an exception. The use of the kernel function improves the accuracy of the DTW which implies that matching frames using the new cost function is a better alternative than employing the euclidean distance to find local optimal warp path only through the constraint window. Combining this approach with the feature extraction method discussed in [9] results in the construction of a model that can applied for **different types** of time series datasets.

Chapter 6

SVD For Time Series Feature Extraction

In the previous chapter, we have explored techniques that can improve the performance of the DTW algorithm in handling high dimensional time series sequences. In this and the following chapters, I will be concentrating in improving the accuracy and run-time of the 1NN classifier equipped with the Euclidean metric on time series sequences. To be precise, I will be exploring feature extraction techniques to extract a smaller set of independent features from the raw sequences that can capture information about the global shape. This formulation will not only achieve dimensionality reduction but will also allow the time series classification to be treated like a normal classification problem.

6.0.3 Motivation

High dimensional data vectors $\{x_n\}$ typically lie close to a non- linear manifold [25] whose intrinsic dimensionality is smaller than that of the input space as a result of strong correlations between the input variables. In data mining, to understand this intrinsic dimensionality of the data, the Single Value Decomposition(SVD) algorithm is widely used. SVD simplifies the data representation by showing only the number of important dimensions that captures the **variance** of the data. Intuitively speaking, SVD projects the high dimensional data vectors to an orthogonal subspace that maximises the variance of the data. The focus of this chapter is to investigate whether SVD can be applied to time

series datasets to extract independent latent features that minimise the run time of the 1NN classifier augmented with the euclidean metric.

6.1 Background

The formal definition of SVD as stated in[18, 19] is given as follows:

Any $M \times N$ real matrix X can be factorized of the form $X = UDV^T$ where

- U is a $M \times M$ orthogonal matrix. The columns of U are the eigenvectors of XX^T
- D is a $M \times N$ rectangular matrix with :

$$D_{ij} = \begin{cases} \sigma^2, & \text{if } j = i \\ 0, & \text{otherwise} \end{cases}$$

  Here D is a diagonal matrix where the diagonal entities are called singular values and are ranked from greatest to least. The number of non-zero singular values determine the rank of the original matrix.

- V is a $N \times N$ orthogonal matrix. The columns of V are the eigenvectors of X^TX. Since X^TX is the covariance matrix, these eigenvectors span the linear subspace that maximises the variance of the data.

Dimensionality reduction is achieved by discarding the eigenvectors in V i.e the columns that are associated with the smallest or zero singular values. Thus, each sample can now be represented by fewer dimensions. For data matrices where M $\ll$ N, there exists at most M-1 singular values. In such situations, considering only the non zero columns in the diagonal matrix D and the associated columns in V achieves dimensionality reduction without any loss of function. The data matrix X can be factorized as $X = U \times D' \times V'^T$ where D is now an M by K diagonal matrix containing only positive values on its diagonal and V′ is the corresponding N by K matrix. The matrix V'^TV captures the full covariance of the data.

6.2 Motivation for applying SVD to time series datasets

Unlike other datasets, clustering/classifying time series sequences by conventional distance metrics, such as the euclidean, may not result in high accuracy. The scope of such metrics is quite limited in the fact that they require sequences to share the same length and comparison of sequences is based on a linear match of the temporal dimensions of the two sequences. This represents a substantial drawback as the metrics fail to take into account that similar patterns may vary in terms of speed, length and scale. Therefore, this provides the motivation to look for preprocessing feature extraction techniques that can extract translation and scale invariant features that capture information about the intrinsic trends in the data.

However, recent works have shown that there exists time series domains where the use of conventional methods can still achieve accurate results. Korn in his paper [19] has shown that for time series sequences that share the same **speed** and **length**, the conventional methods can achieve good clustering/classification performance. If intra class sequences have the same speed i.e no time offsets then they will also share the same time localised trends(see figure 6.1). Applying SVD as a preprocessing step, results in the extraction of latent factors that represent the localised patterns in time. Through mapping the time series sequences into the latent space, we can therefore construct a simplified representation of the data which allows conventional methods such as the euclidean metric to cluster time sequences with lesser run time.

To see how SVD accomplishes this feat, lets consider the following toy example:

Let X be a matrix where rows correspond to customers, columns correspond to days and the values are the dollar amounts spent on phone each day.

X =

customer	wed	thurs	frid	sat	sun
A	1	1	1	0	0
B	2	2	2	0	0
C	1	1	1	0	0
D	5	5	5	0	0
E	0	0	0	2	2
F	0	0	0	3	3
G	0	0	0	1	1

Observation

- There are only two distinct time localised trends. From this we can infer that there are two groups of customers: one group makes calls only during week days while the other group makes calls only during weekends.

Using SVD to factorize the matrix gives:

$$
\begin{aligned}
X &= U \times D \times V^T \\
X &= \begin{pmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{pmatrix} \times \begin{pmatrix} 9.64 & 0 \\ 0 & 5.29 \end{pmatrix} \times \begin{pmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & 0.71 \\ 0 & 0.71 \end{pmatrix}^T
\end{aligned}
$$

Note: The columns in the orthogonal matrix V that are the eigenvectors of the zero singular values in the diagonal matrix have been omitted. Omitting these column vectors results in no loss of information.

Key observations

- The rank of the matrix X is 2. This shows that there are effectively two groups of customers: weekday(business) and weekend(residential) callers.

- The U matrix can be thought of as a customer to pattern similarity matrix. The first column vector of U corresponds to the weekday customers while the second column vector represents the weekend customers
- The V matrix can be thought of as day to pattern similarity matrix. In the case of days, there are two patterns: the weekday pattern i.e {Wed,Thurs,Frid} and the weekend pattern {Sat,Sun}

For SVD to be applied successfully across all time series datasets , there are two fundamental issues that must be addressed:

1. Time complexity: The time complexity of SVD is O(D^3) where D represents the dimensionality of the data. For domains where the dimensionality of the data is really high, applying SVD as a preprocessing step increases the run time of a clustering/classification algorithm.
2. Variable dimensionality: SVD cannot be directly applied to datasets where the individual data points do not share the same dimensionality. This issue is specially seen in time series data sets such as speech. The acoustics samples in such datasets vary in length.

In the rest of this chapter, I will be focussing on addressing the first issue : the problem of high time complexity that is associated with SVD when handling datasets with long time series sequences.

6.3 Improving Time Complexity

For datasets where N(the number of samples) $\ll$ D(dimensionality of the data), Bishop [25] offers the following solution that reduces the time complexity of computing SVD from O(D^3) to O(N^3):

Let X be the constructed D by N data matrix where D is the number dimensions and N is the number of samples and D$\ll$N. By factorizing X, we get X=$U \times D \times V^T$. To achieve dimensionality reduction, we need to consider only the subset of columns vectors of V that are associated with the non-zero singular values. Since N $\ll$D, there will at most N-1 of these singular values. As stated earlier, the columns of the V are the eigenvectors of the covariance matrix XX^T.

The eigen decomposition of the covariance matrix is $XX^T u_i = \lambda_i u_i$. XX^T is D

by D matrix and the computational complexity of computing the eigenvectors is O (D^3).

Now if we multiply both sides by X^T we get :

$$(X^T X)X^T u_i = \lambda_i X^T u_i$$

Setting $v_i = X^T u_i$ gives us the eigenvectors of $X^T X$.

From the above expression, it can be observed that the eigenvectors of $X^T X$ and XX^T share the **same** eigenvalues. Using this insight, we can thus solve the eigenvector problem in spaces of lower dimensionality with computational cost O(N^3) instead of O(D^3). The required eigenvectors of the covariance matrix can then be computed by the using the following expression:

$$u_i = \frac{1}{\lambda^{(\frac{1}{2})}} X v_i$$

For each of the time series datasets used in this project, using Bishop's solution, I have extracted the first k eigenvectors that capture 95% variation of the data with minimal computation cost. By projecting the time series sequences to the orthogonal subspace spanned by the eigenvectors, I have hence constructed a compact representation of each time series sequence. This compact representation in turn allows the 1NN classifier to reduce its time complexity from **O**(ND) to **O**(N^2) where N is number the samples, D is the length of the sequences and $N \ll D$.

6.3.1 Experimental results

To verify the above approach, I have conducted the following experiment :

Objective : The experiment is designed to verify whether applying SVD as a preprocessing step can improve the run time of 1 nearest neighbour classifier on time series datasets, without affecting its accuracy.

Datasets used : InLineSkate and Cinc_ECG_TORSO.

Unlike the speech corpus, the time series sequences of each dataset share the same length. This allows metrics such as the euclidean to compare the similarity between two sequences by conducting a linear match of their temporal

dimensions. Since the number of samples $\ll$ the length of the sequences, the size of the latent space will be smaller than the original input space. For the UCR datasets, I have considered the first k eigenvectors of the covariance matrix that capture **95%** variation of the data as latent features.

Metric Used: Euclidean distance

Methods Compared: 1NN classifier using SVD as a preprocessing step Vs 1NN classifier

Results:

	SVD+1NN classifier	1NN classiifer
Cinc_ECG_TORSO	89.71	89.71
InlineSkate	34.18	34.18

Table 6.1: Accuracy

	SVD+1NN classifier	1NN classiifer
Cinc_ECG_TORSO	0.77s	0.88s
InlineSkate	0.48s	1.03s

Table 6.2: Run times in seconds

Note: The time complexity associated with the Euclidean metric is **O**(N) which is much smaller than **O**(N^2) - the time complexity of the DTW algorithm. Thus, the associated run times associated with the 1NN classifier using the Euclidean metric will be very small in comparison to the run times incurred by the classifier when it employs the DTW metric. Therefore to make effective comparisons of the run times incurred by 1NN classifier equipped with the Euclidean distance metric for embedding different preprocessing techniques, from now on I will be comparing the run times directly and not the ln of the run times.

Observations

- The accuracy of the 1 nearest classifier in both cases is consistent with the 1NN error rate given for each of the datasets in the UCR data source webpage[36].
- Applying SVD as a preprocessing step has no effect on the accuracy on the 1NN classifier but succeeds in reducing the run time of the classifier. Thus using Bishop's[25] formulation, we can use SVD to reduce the time complexity of clustering/classification algorithms for high dimensional time series datasets that satisfy the constraints identified by Korn[19].
- 1NN classifier equipped with Euclidean metric achieves very high accuracy for the Cinc_TORSO_ECG dataset. The similarity of intra class sequences for this dataset can be classified fairly accurately by performing only a linear match of their temporal dimensions. From this observation, we can conclude sequences belonging to the same class tend to share the similar **time localised** i.e time aligned local and global trends.
- Using the euclidean metric, the 1 NN classifier is seen to achieve very low accuracy for InlineSkate dataset.This shows that unlike the Cinc_TORSO_ECG datasets, the intra class sequences are not aligned with respect to the time axis.

Chapter 7

Improving The Accuracy Of The Euclidean Metric

In the last chapter, we have seen that the 1 nearest neighbour classifier combined using the Euclidean metric achieves good performance in both accuracy and run time in classifying sequences belonging to the Cinc_ECG_TORSO if applies single value decomposition as a preprocessing step. The Cinc_ECG_TORSO is a rare example of a time series data set where the local and global trends of intra class sequences are nearly aligned with respect to the time axis(see figure 7.2). This allows sequences to be fairly accurately classified based only on a linear match of their temporal dimensions.

For most time series data sets however, conducting linear matches of time series sequences presents a limitation. The intra class variances will in general be very high as sequences of the same class may vary in size, shape or speed. The InlineSkate dataset is a prime example of such a dataset. Figure 7.1 shows examples of two instances of class 2 in the InlineSkate dataset. The two instances share the same global trend but posses different local trends: in the period between t= 400 to 600, the first sequence is experiencing an upward trend while the second sequence is seen to experience a downward trend. Performing comparison through only a linear match of the temporal dimensions is inappropriate in such cases as the differences in the local trends increase the degree of dissimilarity between the two patterns. This creates the motivation to investigate feature extraction methodologies that map the sequences to a an appropriate feature space that captures information about common class attributes such as global shape, trends etc. In doing so we

can minimise the intra class variance of time series sequences and allow the time series classification problem to be approached as a general classification problem.

The focus of this chapter is to explore feature extraction techniques that can capture information about the **global shape** of time series sequences. In the first half of this chapter, I will be exploring wavelet-based feature extraction techniques and will subsequently look into how they can be used alongside SVD to improve the accuracy of the 1 nearest neighbour classifier for datasets that share the same time localised trends. While, in the later half of this chapter, I will be investigating further feature extraction methodologies that can be combined with SVD and the wavelet transform to capture information about the shape of the dominant trends of sequences. This is intended to improve the accuracy of the 1NN classifier using the Euclidean metric for time series datasets where intra class sequences share global trends that are not aligned with respect to the time axis.

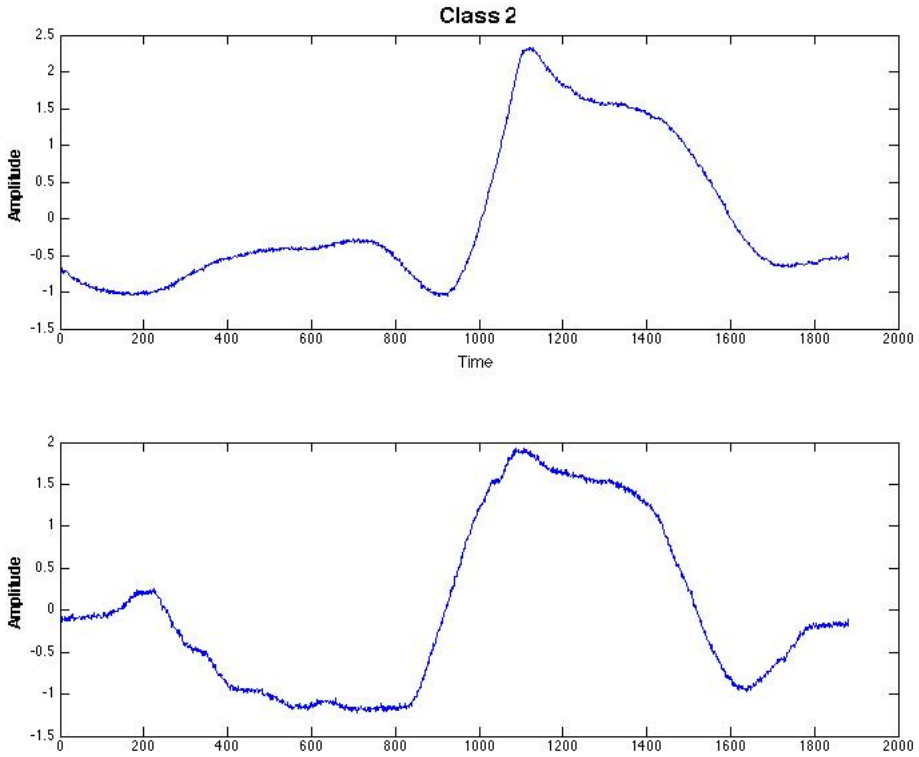


Figure 7.1: Time series sequences corresponding to class '2' in the InlineSkate dataset

7.1 Wavelet-based Feature Extraction

7.1.1 Background

The inspiration for the wavelet transform came from the idea of multiresolution analysis[26]. A multiresolution analysis decomposes a signal into a smoothed version of the original signal and a set of detail information at different scales. This type of decomposition is most easily understood by thinking of a picture (can be thought of as a two dimensional signal) and removing from the picture information that distinguishes the sharpest edges, leaving a new image that is slightly blurred. This blurred version of the original picture is a rendering at a slightly coarser scale. We then recursively repeat the procedure on the smoothed signal. The wavelet transform specifies such a multiresolution decomposition, with the wavelet functions defining the bandpass filter that determines the detail information and associated with each wavelet is a smoothing function, which defines the complementary lowpass filter.

Mathematically, we can motivate the definition of the wavelet transform as follows:

Let $L^2(R)$ be the set of all discrete functions with finite energy. Any discrete function $f \in L^2(R)$, can be written as a linear combination of translations of the scaling function $\phi(st)$ as shown below:

$f(t) = \sum_{n=0}^{N} \alpha_n \phi(st-n)$ where $\phi(st)$ is defined as :

$$\phi(st) = \begin{cases} 1 & \text{if } 0 \leq st < 1 \\ 0 & \text{otherwise} \end{cases}$$

Under this representation, the following facts hold.

i The scaling function is orthogonal to its integer translates i.e

$$< \phi(st-k), \phi(st-j) >= \begin{cases} 1 & \text{if } j == k \\ 0 & \text{otherwise} \end{cases}$$

ii As we increase the scale, the support of the function decreases.

iii The subspaces spanned by the scaling function at low scales are nested

within those spanned at higher scales. For example if represent s as powers of 2 then: $\phi_{-1}(t) = \phi(2^{-1}t) = \phi_0(t) + \phi_0(t-1)$.

Before proceeding any further, it is necessary to define the concept of the wavelet. A wavelet [27, 28, 29] is a smooth and quickly oscillating function with good localisation in both frequency and time. To perform wavelet transformation the Haar function is commonly chosen as the mother wavelet ψ. For any scale s, the function $\psi(st)$ is orthogonal to $\phi(st)$. But like $\phi(st)$, $\psi(st)$ can be expressed in terms of higher order scaling functions $\phi(s't)$ where $s' > s$. Using this knowledge alongside (iii), the wavelet transformation is applied to decompose a finite signal f by projecting f onto two orthogonal subspaces D_1 and $A_1$of $L^2(R)$ where $L^2(R) = A_1 \oplus D_1$. If $f(t) = \sum_{n=0}^{N} \alpha_n \phi(st-n)$ then D_1 is spanned by $\{\psi(s't-n)|n \in [1,..N]\}$ and A_1 is spanned by $\{\phi(s't)|n \in [1,,N]\}$ where $s' < s$.

Note :To ensure that $\{\psi(s't-n)|n \in [1,..N]\}$ forms an orthogonal basis, the scales are restricted to negative powers of 2[30, 27].

The process is recursive. After removing the first set of detail information from f, we are left with a slightly smoothed version of f. We iteratively remove detail information by applying the transform on the smoothed versions at each scale to progressively extract coarser details of f . We stop the process once we have enough detail information or a smooth enough version to do the analysis we desire. Therefore for given any signal X, the wavelet coefficients $H_j(X)$ at any scale j can be represented by the series of coefficients $\{A'_j, D'_j, D'_{j-1}...D'_1\}$. A'_j represents the coefficients of the scaling function at scale j or in other words it corresponds to the coordinates of the signal when it projected in the subspace A_j. Similarly, $D'_j...D'_1$ represent the coefficients of Haar wavelet at different scales or in simpler terms, they correspond to the coordinates of the signal when it is projected to each of the orthogonal subspaces $D_j...,D_1$. To see how this works, lets consider the following example:

Let $f(x)$ be a discrete signal given by $f(x) = \{9,7,3.5\}$. This function can be represented as linear combination of the scaling function $\phi(st)$ where s=1. As we have seen in the previous sub section, the wavelet decomposition is a recursive process where at each level j, we project the signal in the two orthogonal subspaces A_j and D_j. Since A_j is constructed using the basis $\{\phi(2^{-j}t-n)|n \in [1,..N]\}$ and D_j is constructed using the basis $\{\psi(2^{-j}t-n)|n \in [1,..N]\}$, the wavelet transform can hence be seen as a series of averaging

and differencing operations on a discrete time function. We compute the average and difference between every two adjacent values of the discrete signal $f(x)$. The recursive procedure to find the wavelet transform of a discrete function is shown below.

Decomposition Level	Averages	Coefficients
0	(9 7 3 5)	
1	(8 4)	(1 -1)
2	(6)	(2)

The level 0 is the full resolution of the discrete function. At the first level of decomposition, (8 4) are obtained by taking the average of (9 7) and (3 5) and (1 -1) are the differences of (9 7) and (3 5) divided by two respectively. The procedure is repeated on the approximated coefficients (8 4) to obtain (6) and (2) in the next level.

The subspaces $\{A_m\}|_{m=1}^{M}$ hence satisfy $A_{j+1} \subset A_j$. The details of f at any scale **m** is a projection of f onto the subspace D_M and is captured by the coefficients l D'_M. The projection can be represented by the map $Q_M : L^2(R) \rightarrow D_M$ where $Q_M f = \sum_{n=1}^{K} < f, \psi_{Mn} > \psi_{Mn}$(the subspaces D_M are spanned by dilations and translations of the mother wavelet ψ).

Furthermore there exists another operator P_M that maps f to its approximation at scale **m**: $P_M : L^2(R) \rightarrow V_M \subseteq L^2(R)$. The functional space $L^2(R)$ can therefore be expressed as $L^2(R) = \oplus_{m=1}^{M} D_m \oplus A_M$. where any finite signal $f \in L^2(R)$ can be expressed as :

$$f(t) = \sum_{n=0}^{N} \alpha_n \phi(t-n) = P_M f + \sum_{m=1}^{M} Q_m f$$

7.1.2 Wavelet-based Features

The wavelet transformation allows a time series sequence to be described in terms of an approximation of the original sequence plus a set of details ranging from fine to coarse. Thus, for given any time series X, the wavelet coefficients $H_j(X)$ at any scale j can be represented by the series $\{A'_j, D'_j, D'_{j-1}...D'_1\}$ where A'_j are the coefficients of the scaling function at scale j and $D'_j...D'_1$ are the coefficients of Haar wavelet at different scales. The se-

quence A'_j corresponds to an approximation(smoothed version) of the signal at scale j. The coefficients A'_j are the **amplitude** values of the smoothed signal. These coefficients can be considered as global trend descriptors when intra class sequences share the same time aligned global trends. The **localised** changes o the other hand are captured in the series of coefficients $\{D'_j...,D'_1\}$[28] where the level of details captured at each D_j becomes coarser as we move to lower scales. The decomposition presents **no loss** of information. Since the wavelet transform is a one to one linear map, the original signal can be fully reconstructed from the approximation part and the detail parts.

The wavelet transformation achieves both time and frequency localisation of a time series. This allows us to analyse singularities specific to both times and frequency bands. The singularities at different frequency ranges are captured by coefficients of the details and the approximation signals at different scales. Whereas for signals that are nonzero only during finite spans of time, the wavelet transform has nonzero elements that are concentrated around that time for all scales. This allows us to analyse singularities localised in different time intervals. Thus as a feature extraction step, performing wavelet decomposition can be highly advantageous as it allows us to decompose a signal and provide a richer description of its behaviour.

As we have discussed in the previous section, the intra class sequences in the Cinc_ECG_TORSO dataset share similar local and global trends. In such problem domains, the coefficients of both ψ and ϕ at different scales are required to form accurate comparison. Figure 7.2 show examples of two instances belonging to class '3' and two instances belonging to class '4'. From an observation of the plots, it can been seen that apart from sharing global trends, intra class sequences also tend to share similar local trends. Information about the global shape is captured by the coefficients of ϕ which behave as global trend descriptors while the finer details are expressed by the coefficients of ψ. Thus, to make accurate comparison, it is necessary to take the coefficients of both ψ and ϕ into account.

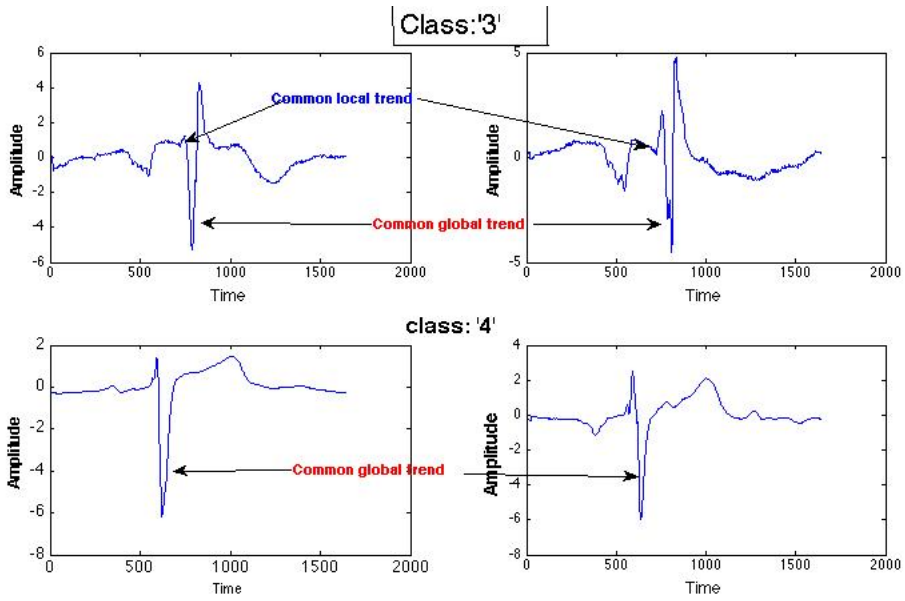


Figure 7.2: The top two plots corresponds to instances of class '3' and the bottom two plots correspond to instances of class '4' of the Cinc_ECG_TORSO dataset

However, for the the InlineSkate dataset, the global shape is the only commonality between sequences in the same class. Here intra class sequences share only global similar trends. Figure 7.3 shows examples of samples belonging to the class '2'. The red lines represent smoothed curves fitted to the time sequences to to capture the overall shape. Both instances share trends that are roughly cubic in nature.

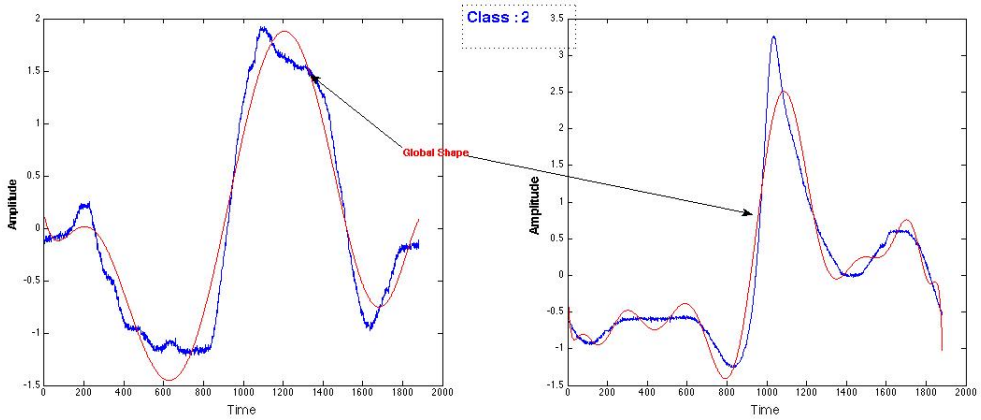


Figure 7.3: The plots correspond to instances of class '2' of the InlineSkate dataset

In such cases, it can be assumed that the coefficients of ϕ play a greater role in discriminating classes than the coefficients of the wavelets.

To investigate whether applying wavelet decomposition prior to SVD as preprocessing step improves the performance of the 1 nearest neighbour, I conducted the following experiment:

Datasets: Cinc_ECG_TORSO and InlineSkate

Distance metric used: Euclidean distance

Preprocessing: Wavelet decomposition followed by the single value decomposition procedure discussed in 6.3.

1. Step 1: Feature Extraction by applying wavelet decomposition

   Cinc_ECG_TORSO dataset: To capture information about both local and global trends with respect to various frequencies and time, I have applied wavelet decomposition up to scale 17. Through experimental testing, I have found that stopping the recursive decomposition at scale 17 is enough to capture the all the information about the behaviour of global and local trends at different frequency bands.

   Thus, each sequence X is decomposed into the series of coefficients $\{A'_{17}, D'_{17}.....D'_1\}$ where A'_{17} represents the amplitudes of the smoothed signal and D'_j represents the amplitudes of Haar wavelet at scale j. The coefficients of D'_j capture the time localised behaviour of the signal at the frequency band associated with scale j. Since the wavelet transform is a linear one to one mapping, each sequence X is hence mapped to a series of subsequences $\{A'_{17}, D'_{17}.....D'_1\}$ where the sum of the length of the subsequences is equal to the length of X . The series can be thought of as a sequence of discrete orthogonal signals where the signal corresponding to index j is constructed by composing the integer translates of the Haar wavelet at scale j i.e $\sum_{n=1}^{K} < f, \psi_{jn} > \psi_{jn}$ with the exception of A'_{17}.

   For this analysis, I have omitted the detail coefficients corresponding to scale 1 i.e D_1. The reason being regions in high frequency tend to have a lot of noise embedded in them. Since the first level details capture information about the behaviour of the signal at the highest frequency bands, discarding those details achieves reduction in both the noise and the dimensionality of the sequence. Since all sequences in the dataset share the same length and can be decomposed into the series $\{A'_{17}, D'_{17}.....D'_1\}$, discarding D_1 reduces the dimension of the time series sequences by half.

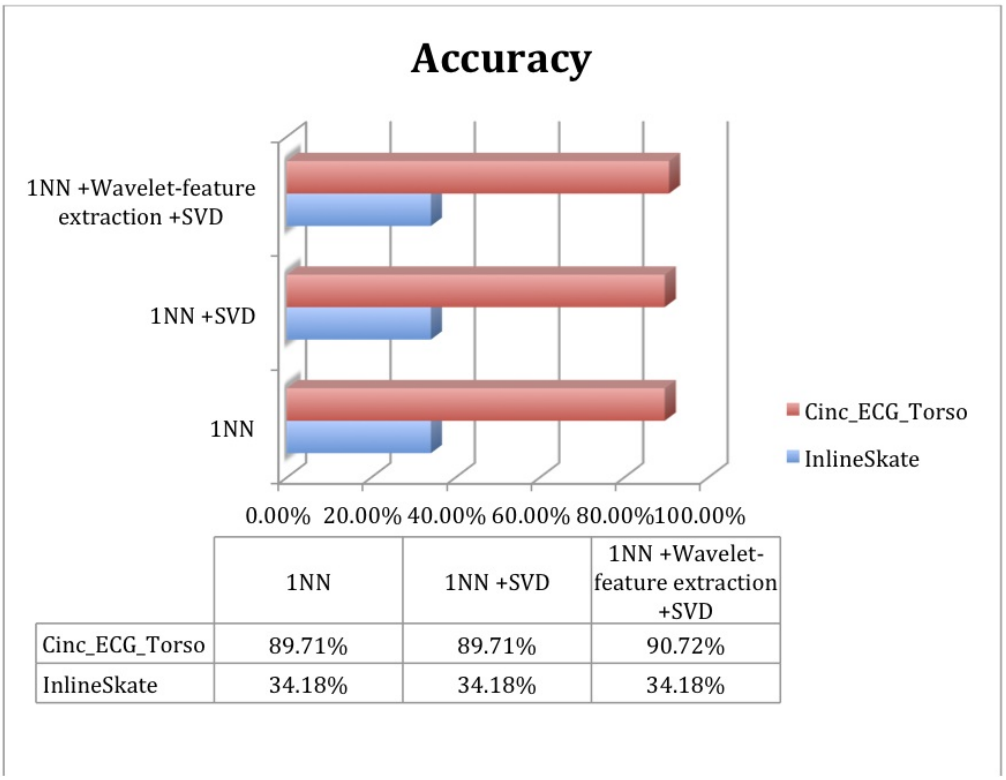
InLineSkate dataset: For this dataset, I have applied wavelet decomposition up to a depth of 4 levels. Since the commonality of the classes is captured by the global shape. I have only considered the approximation coefficients ϕ at scale 4 i.e. the coefficients A'_4 .

2. Step 2: Extracting latent features from wavelet coefficients

   As I have mentioned in the previous section, the wavelet transform can be seen as a series of averaging and differencing operations on a discrete time function. At the first level, the discrete function is decomposed into two subsequences $\{A_1\}$ and $\{D_1\}$ of equal length. The coefficients of A_1 represent the amplitudes of the smoothed signal obtained by averaging every two adjacent values of the original signal. The process is recursive and decomposition is applied again on the approximated signal to separate even more coarser details . Thus after level j of the wavelet decomposition, we obtain a smooth signal that is 2^{-j} smaller than the original sequence.

   For the InlineSkate and Cinc_ECG_TORSO datasets, the sequences within each dataset share the same length. Thus the resultant data matrix constructed using the coefficients extracted from the wavelet decomposition will be full but however in the case of Cinc_ECG_TORSO dataset, the number of columns(this corresponds to the dimension of the data) will be reduced to half, while for the InlineSkate dataset, the number of columns will reduced to 2^{-4}. SVD is then applied to each of the data matrix to extract latent features that capture 95% variation of the data in the constructed wavelet feature space.

The performance of 1NN classifier using SVD and wavelet transform as preprocessing step was compared with the performance of 1NN classifier using SVD as a preprocessing step and the baseline 1NN classifier. A summary of the results are as follows:



	1NN	1NN +SVD	1NN +Wavelet-feature extraction +SVD
Cinc_ECG_Torso	89.71%	89.71%	90.72%
InlineSkate	34.18%	34.18%	34.18%

Figure 7.4: Accuracy

Dataset	1NN	1NN+ SVD	1NN+ Wavelet-feature extraction +SVD
Cinc_ECG_TORSO	0.88s	0.77s	0.74s
InlineSkate	1.03s	0.48s	0.45s

Table 7.1: Run time in seconds

Observation

- As evident from the run time results, applying the wavelet feature extraction as prior step to SVD reduces the run time of the 1NN classifier even further.
- For the Cinc_ECG_TORSO data set, removing the detail coefficients of scale 1 i.e D_1 leads to a fractional improvement in the accuracy of the 1 nearest neighbour classifier. From this observation, we can deduce two facts:
    1. Because the accuracy has not decreased, we can conclude that the

information contained in the finer details of sequences present redundant information.

2. Removing such information has the affect of decreasing the intra class variance as evident from the increase in accuracy.

- In the case of the InlineSkate data set, considering only the approximation part of signal i.e A_4 results in no decrement in the accuracy of the 1 NN classifier. This shows that when comparing sequences using the euclidean metric, the details in the first 3 levels: D_3, D_2 and D_1 play no role in the classification process. To investigate whether applying further wavelet decomposition can improve the accuracy of the classifier, I repeated the experiment for the InlineSkate dataset again but this time I have considered the coefficients of A_5(the approximation of A_4)and A_6(the approximation of A_5). A summary of the results is given below:

Dataset	Using A_6	Using A_5	Using A_4
InlineSkate	33.82 %	34.18%	34.18%

Table 7.2: Accuracy results

  - Discarding coarser details from the approximated signal A_4 decreases the accuracy of the 1NN classifier as we increase the number of decompositions. This shows that for this dataset, we cant achieve any improvement in the accuracy of the 1NN classifier by considering approximations from further decompositions using the wavelet transform.

The subsequence A_j as I have mentioned before corresponds to an approximation(smoothed version) of the signal at scale j. The coefficients of A_j are the amplitude values of the smoothed signal. These coefficients are constructed by averaging every two adjacent values of the approximated signal corresponding to j-1 and can be considered as global trend descriptors when intra class sequences share the **same** time aligned global trends.

From the analysis above, replacing the original sequences with the smaller approximated sequences A_j doesn't improve the accuracy of the 1NN classifier for the InlineSkate dataset. And in fact, the accuracy of the classifier decreases when we consider approximations from the lower scales i.e j $>$4. Thus, we

can conclude that the global trends shared by intra class sequences are not localised with in the **same** intervals of time. Metrics such as the euclidean will be hence inept in utilising the embedded information about the global shape as they are constrained to conducting only linear matches between temporal dimensions of sequences

However, for datasets in which intra class sequences share the same time localised/aligned global trends, we can expect that by applying the wavelet decomposition as a preprocessing step and considering the approximated part of the signal at a scale j, the accuracy of the baseline 1NN classifier can be improved. To verify this assertion, I performed an additional experiment on the UCR dataset 'Synthetic Control'. This dataset have been chosen because experiments have shown[9] that the intra class sequences of the dataset are linked with each other through time aligned global trends.

Preprocessing Steps

The dimension i.e the length of the time series sequences is 60 which is very small. Hence the wavelet decomposition has been applied only up to level 1 i.e only the coefficients of A_1 are considered. The application of SVD was deemed not necessary for this dataset since the sequences have very small lengths.

The performance of 1NN classifier using wavelet transform as a preprocessing step was compared with the performance of the baseline 1NN classifier and the performance of the 1NN classifier combined with the DTW metric. A summary of the results are as follows:

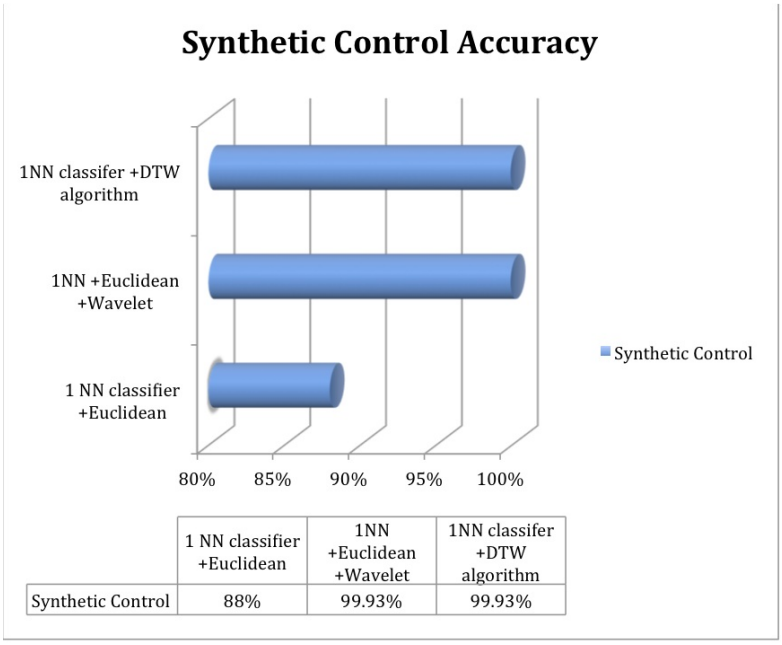


Figure 7.5: Accuracy

Dataset	1NN +Euclidean	1 NN +Euclidean + Wavelet	1 NN +DTW
Synthetic control	0.32s	0.30s	777.3s

Table 7.3: The run time

Observation

- Removing unnecessary details by smoothing the signals improves the accuracy of the 1NN classifier by 10%. The resultant accuracy is identical to the best accuracy achieved by combining 1NN classifier with the DTW algorithm. Furthermore, the run time of 1NN classifier using the Euclidean metric is 10^4 times smaller than the run time of the 1NN classifier combined with the DTW metric.

  This proves that by applying the wavelet transform as a preprocessing step, the 1NN equipped with the Euclidean metric can identify intra class sequences that roughly share the same time aligned global trends with high accuracy without being subjected to high run times.

7.2 Curvature-based Feature extraction

In the previous section, we have observed that for datasets such as the InlineSkate dataset, the main commonality between intra class sequences is the global shape. Applying wavelet decomposition and considering only the approximation part of the signal A_4 allows us to achieve dimensionality reduction but without any improvement in accuracy. The low accuracy achieved by the classifier is primarily due to the limitations incurred in comparing sequences using the euclidean metric. The individual amplitude values on their own are not reflective of the global shape of the sequence. This creates the motivation to explore feature extraction methodologies that capture information about the shape of the sequence.

To address this issue, I propose the following 4 step feature extraction methodology to extract useful features from the InlineSkate dataset:

i Apply wavelet decomposition up to scale j and consider only the approximation part of the signal A_j. This achieves dimensionality reduction by pruning away unwanted details while preserving information about global trends.

ii Replace the amplitude value associated with each point of the approximated signal with curvature value of the smoothed signal associated with the corresponding point(details to follow).

iii Apply fourier transform and consider only the first half of the fourier coefficients(details to follow).

Note: This step should only considered when we are working with time series datasets that have very high dimensionality.

iv If the dimension/length of the sequences $\gg$ than the number of samples after the first 3 steps then apply SVD on the result sequence of fourier feature descriptors to extract a smaller set of latent vectors that capture the 95% variation of the feature vectors in the fourier feature space.

In the following sections, I provide a detail description of the procedures specified in step (ii) and step (iii) of my proposed methodology.

7.2.1 Motivation for curvature based features

Replacing the individual amplitude values with the curvature estimates of the corresponding points can allow the euclidean metric to take into account the **shape** of the signals at the corresponding points when conducting a linear match. The curvature of a geometric object is a quantitive description of the shape of that object[32]. It dictates the amount by which a geometric object deviates from being flat or straight. Formally, the curvature of a curve can be defined as follows:

Let γ be a differentiable curve on $\mathbf{E}^3$(Euclidean Space) such that

$$\gamma : (-\epsilon, \epsilon) \rightarrow U \subseteq \mathbf{E}^3$$

$$\gamma(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$$

The derivative of this function $\gamma'(t) = \begin{pmatrix} x'(t) \\ y'(t) \\ z'(t) \end{pmatrix}$ is the velocity of this curve.

Thus by assuming that this curve is parameterised by arc length (i.e. the curve is parameterised to have unit velocity $\|\gamma'(t)\| = 1$) we define the *curvature* of as curve as

$$\kappa = \|\gamma''(t)\|$$

which in fact is none other than the magnitude of the acceleration of a curve.

Geometrically one can imagine κ as a parameter determines the amount of the bending of that the curve experiences in the flat euclidean space.

7.2.2 Extracting curvature based features

To extract features that capture the **shape** of the dominant trends in a sequence, I performed the following two step feature extraction process.

Step 1: Extraction of curvature estimates

To compute the curvature estimates at each point in a time series sequence, I have fitted a **smoothed cubic spline** to each sequence. A spline is a polynomial

function that is piecewise-defined, and possesses a high degree of smoothness at the places where the polynomial pieces connect [33]. A cubic spline is a specific type of spline which is constructed using piecewise polynomials of order 3.

Although applying the wavelet decomposition reduces the degree of noise, fitting a cubic spline directly on the sequences will still not be ideal as the approximated signals may still have jagged regions as shown by figure 7.6. The figure shows the approximated signal and its associated fitted curve of an instance in the InlineSkate dataset. To overcome this issue, I have thus applied smoothed cubic spline to fit smooth curve to each sequence(given by the blue plot in figure 7.6) .

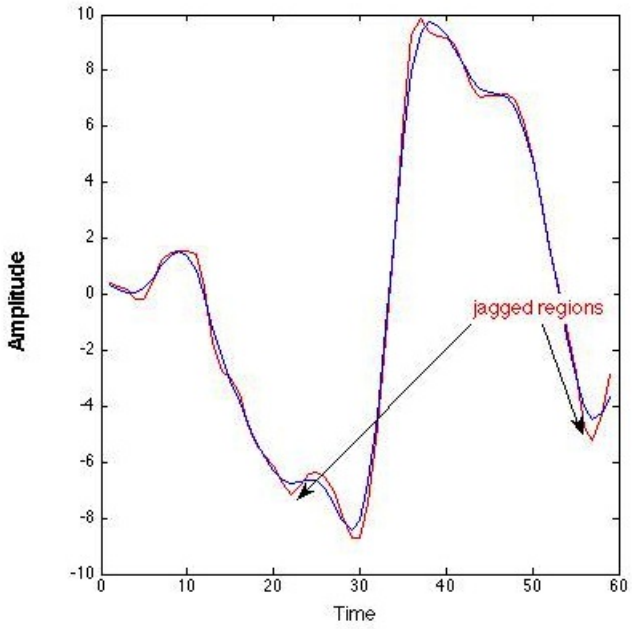


Figure 7.6: The red plot corresponds to the approximated signal A_4 and the blue plot corresponds to the fitted cubic spline

Mathematically, the smoothing spline estimate $\hat{g}(x)$ of the actual function $g(x)$ is defined to be the minimiser of :

$$\lambda \sum_{i=1}^{n} \{y_i - \hat{g}(x_i)\}^2 + (1-\lambda) \int_a^b \{\hat{g}''(t)\}^2 dt$$

where $\lambda \in [0,1]$

- As $\lambda \to 1$,the smoothing spline converges to the interpolating spline.

- As $\lambda \to 0$,the roughness penalty becomes paramount and the estimate converges to a linear least squares estimate.

The critical parameter here is λ and it's value determines the nature of the fit. For the sequences in the InLineSkate dataset, exact fitting is unnecessary as the sequences have jagged regions in them. Ideally, we want to choose a value of λ that allows the estimated curvatures to directly correlate with significant events/trends in the time series sequence.

For my analysis, I have chosen the value of 0.2. Through conducting experiments, I have observed that setting λ to this value allows the computed curvatures to reflect significant events in the time series sequence. An illustration of this can be seen in figure 7.7. The plots correspond to an instance of class '7' from the InlineSkate dataset. The left hand plot corresponds to the approximated signal A_4 extracted through wavelet decomposition while the righthand plot represents the curvatures computed at each of the time stamps. From the observation of the figure, it can be seen that the computed curvatures do reflect information about significant events in the time series sequence. For instance the peak value on the curvature plot at t=148 is in accordance with the sharp bend of the signal at that time.

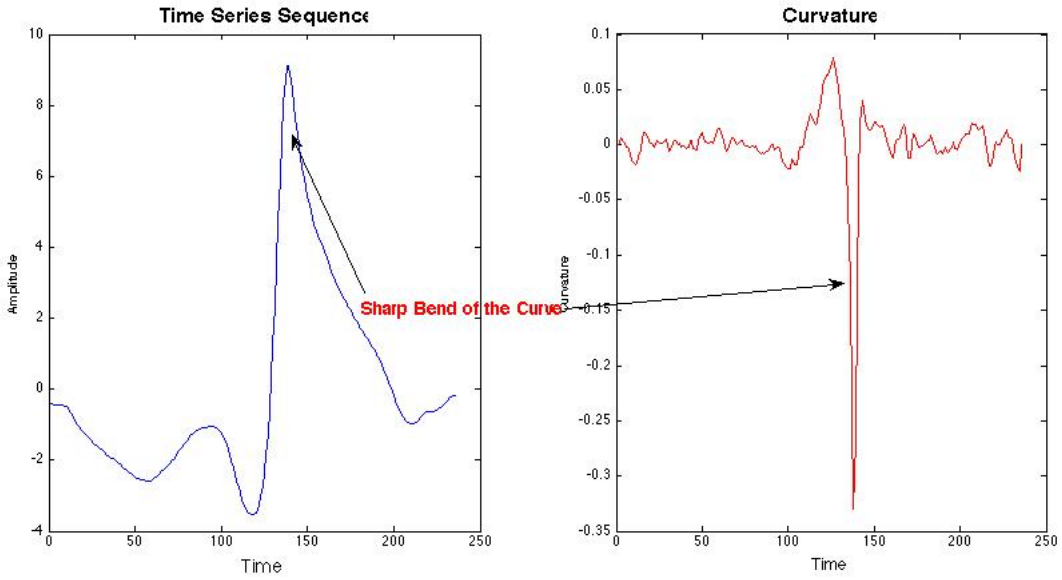


Figure 7.7: The left hand plot represents the time series sequence while the right plot corresponds to curvatures computed at each point in the time series sequence

The curvatures κ at each point of the fitted smooth cublic curve is computed

using the following expression:

$$\kappa = \frac{\hat{g}''(x)}{(1 + \hat{g}'(x)^2)^{\frac{3}{2}}}$$

Step 2: Extract features that reflect the global shape

The main objective here is to extract features that reflect the **global** shape of the time series sequence. Although applying the wavelet decomposition reduces the dimension (i.e the length) to $\frac{1}{16}$ of the original length, the resultant sequences especially in the case of **high** dimensional time series sequences may still be long enough to retain some local trends. The above mechanism embeds the information of local curvatures to each point of the time series sequence. Hence for datasets where intra class sequences are connected by just by global trends, the curvature associated local trends may increase the intra class variance.

What we are really interested in is in the extraction of global shape descriptors. The shapes of the global trends of a signal are preserved in the low frequency bands while the higher frequency bands encode information of local shapes [34]. Thus, to extract global shape descriptors, I have applied the discrete fourier transform[35] on the extracted curvature sequences.

The Fourier transform maps time into frequency and phase. For each frequency the fourier transform yields an amplitude and a phase value. Thus, the transform captures the behaviour of the signal at different frequency band. Since the amplitudes represent curvature values estimated at step 1, the fourier transform therefore capture the behaviour of the **shape** of the signal at different **frequency** bands. The curvature-sequences can now be represented as the sum of sine and cosine waves whose phase and 'amplitude' are given by the fourier transform.

For any time series sequence, the coefficients of the discrete fourier transform for each frequency band is computed as follows:

$$X_k = \sum_{i=1}^{n} x_n . e^{-2\pi i k \frac{n}{N}}$$

To extract global shape descriptors only the magnitudes of the fourier coefficients have been used, Since global trends [27, 34] are preserved in the lower

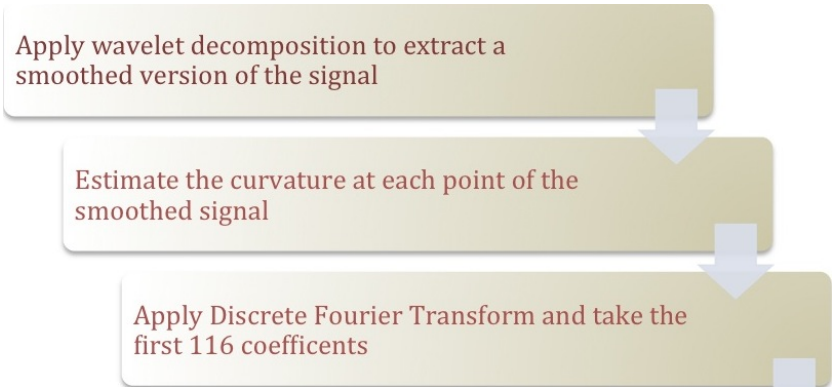
bands, only the first few fourier coefficients have been considered. For the InlineSkate dataset, I have only selected the first 116 coefficients as features for nearest neighbour classification. Since the number of fourier coefficients is smaller than the number of samples, for this dataset, the 4th step i.e SVD was deemed not necessary.

To investigate whether using the 3 step feature extraction method improves the accuracy of the 1 nearest neighbour classifier, I re-ran the experiment again on the InlineSkate dataset but using the proposed feature extraction method as a prior preprocessing step. The details of the conducted experiment are as follows:

7.2.3 Experiments

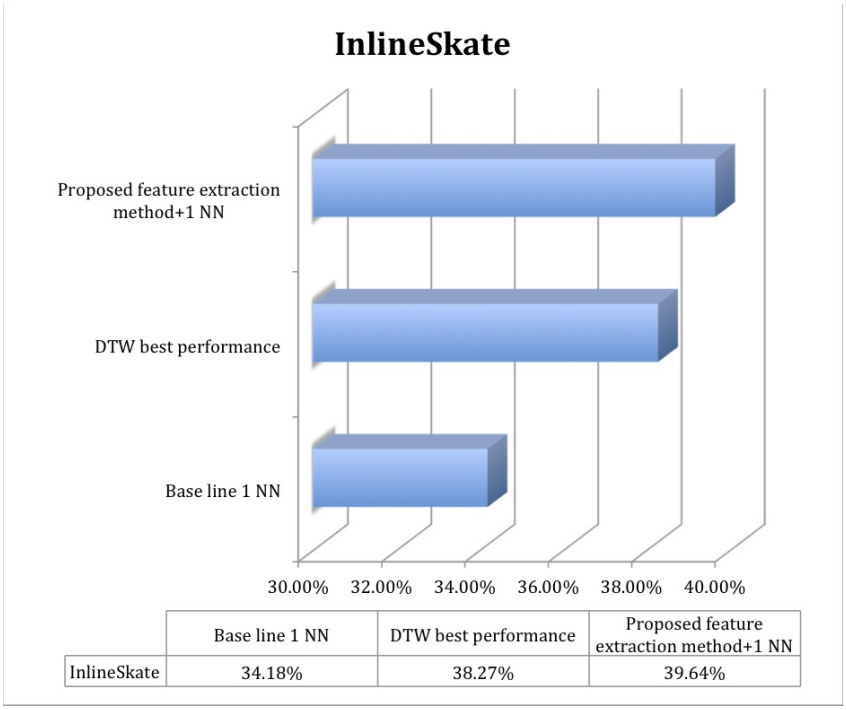
Dataset used: InLineSkate dataset

Preprocessing step



Note: Dimensionality reduction occurs at step 1,3 and 4. Redundant features are pruned away at this steps to finally result a small set of latent factors that be used to cluster/classify inter class sequences that are distinguished by global trends.

The performance of 1NN classifier using 4 step feature extraction method as a preprocessing step was compared with the performance of the baseline 1NN classifier and the performance of the 1NN classifier combined with the DTW metric. A summary of the results are as follows:



	Base line 1 NN	DTW best performance	Proposed feature extraction method+1 NN
InlineSkate	34.18%	38.27%	39.64%

Figure 7.8: Accuracy

Dataset	1NN +Euclidean distance	1NN + Euclidean distance +proposed feature extraction method	1NN +DTW metric
InlineSkate	1.03s	0.199s	3.290×10^6

Table 7.4: Run time

Observation

- Embedding the baseline 1NN classifier with the proposed feature extraction method reduces the run time of the the baseline classifier by an order of 9.
- The global shape features extracted from the proposed feature extraction method are more successful in capturing information about the global shape than the raw amplitude values. From the table above, it can be seen that the 1NN classifier's accuracy has improved by 5.46% when its employs the 3 step feature extraction method as a preprocessing step in comparison to using the raw values directly.
- Interestingly, for this dataset, conducting a linear match of the latent features allows the the 1 NN classifier to achieve accuracy identical to the accuracy achieved when it employs the DTW algorithm as a

similarity metric. Furthermore, the run time of 1NN classifier using the Euclidean metric is 10^7 times smaller than the run time of the 1NN classifier combined with the DTW metric.

As we have discussed in the previous section, for time series sequences, the DTW algorithm makes a richer comparison than the euclidean metric. By warping the time axis, the DTW algorithm compares each point in one sequence with points at different temporal regions in the second sequence whereas the euclidean metric is only restricted to conducting linear matches between the temporal dimensions. However from the experimental results, it can be seen that for the InlineSkate dataset, if we embed the proposed 3 step preprocessing step then the performance of using euclidean metric is equivalent to applying the DTW algorithm on the raw sequences.

To verify that the proposed feature extraction method is not tailored for this particular time series dataset, I conducted a separate experiment on the UCR dataset CBF[36] dataset. The CBF dataset have been chosen because experiments have shown[9] that the intra class sequences of the dataset tend share the same global trends.

Setup:

- From examining the plots of the time series sequences, I have applied wavelet decomposition only up to level 3 and taken the coefficients of the approximated signal A_3.
- The value of λ chosen for smoothed cubic spline fitting is 0.35. At this value, the classifier is seen to yield the best results.
- Since the length of the sequences is only 128, the information of individual local trends is almost pruned away in the early levels of decomposition. Figure 7.9 shows the plot of a raw signal corresponding to an instance and the signal constructed from A_1 using the reverse wavelet transform [27].

  Thus when we repeat the wavelet decomposition 3 times and take A_3, we can expect the coefficients of A_3 to correspond to a smoothed signal where all the local trends are pruned away. The length of the A_3 is now 8 times smaller than the original sequence, therefore, the fourier extraction step is deemed not necessary in this case as the extracted sequences are very small in length.

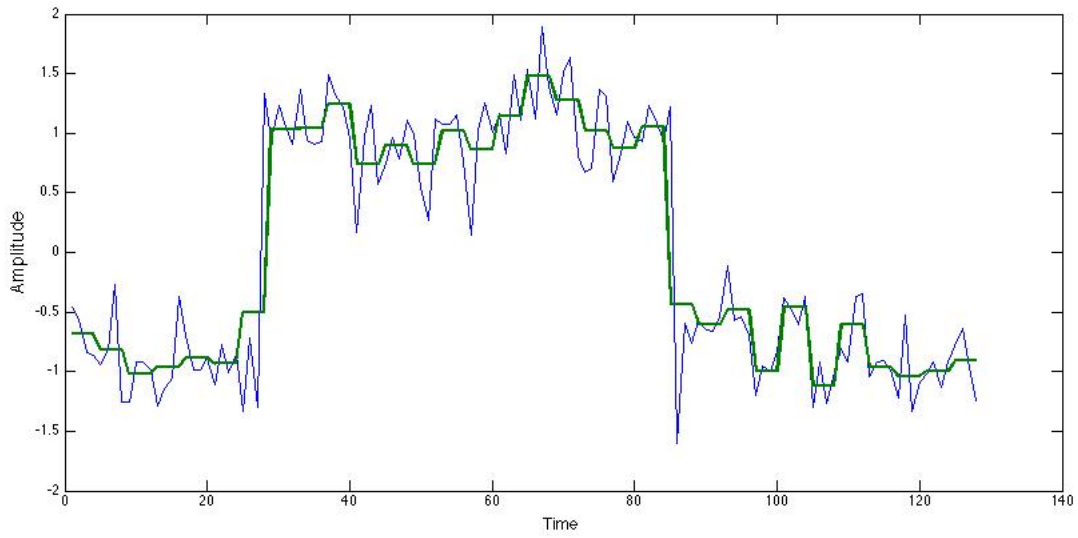


Figure 7.9: The green plot corresponds to signal constructed from A_1 and the blue plot presents the original signal

The accuracy of the model was compared against the accuracy of the baseline 1NN classifier and the best accuracy of 1NN classifier using the DTW algorithm[9].

A summary of the results are as follows:

Dataset	baseline 1NN	1NN + proposed method	1NN + DTW algorithm
CBF	85.2%	95.78%	99.92%

Table 7.5: Accuracy results

Dataset	baseline 1NN	1NN + proposed method	1NN + DTW algorithm
CBF	0.109s	0.094s	1048.2s

Table 7.6: Run time results

Observation

- Augmenting the baseline 1NN classifier with the proposed multi-step feature extraction process improves both the run time and accuracy of

the algorithm, The classifier achieves an improvement of over 10% in accuracy and incurs a run time which 9 times smaller than

- By using the features extracted from the proposed multi step feature extraction method, the accuracy of the 1NN classifier has improved by over 10%. The results show that the proposed method allows the extraction of useful latent features that capture information about the global shape for time sequences distinguished by their global trends.

7.3 Conclusion

To summarise, in this chapter we investigated methods to improve the accuracy of 1NN classifier when it employs the Euclidean metric on time series datasets. In section 7.1, we have observed that applying the wavelet transform as a preprocessing step to extract coefficients can improve the identification of intra class sequences that roughly share the same time-aligned local or/and global trends. Whereas in the later chapters, we have observed that for time series datasets where intra sequences share a global shape, applying the preprocessing strategy proposed in section 7.2 can help the euclidean metric to be as equally effective as a DTW metric when classifying sequences without being subjected to high run times.

Chapter 8

Things to do

- apply the proposed preprocessing method on the Tidigits data sets . However there are some issues:
  1. sequences are not of the same length probable fix : i) rescale the sequences to share the same length

     ii)embed 0 at the end of each sequence to make as long as the longest sequence
  2. the frequency is too high..hard to capture the shape of the wave due to much varation Prob fix: translate the problem to a 2D problem. Map each signal to an image and compute the shape of the image using the 4 steps as before
- Discussion—need to discuss with you on that
- If i get time which I doubt i want to try to the test the 4 step preprocessing methodology on a different UCR dataset.

Bibliography

[1] R. Gautam Das, King ip Lin, Mannila, H., "Rule Discovery From Time Series," *4th Int'l Conference on Knowledge Discovery and Data*, 1998.

[2] U. Fayyad, C. Reina, and P. S. Bradley, "Initialization of Iterative Refinement Clustering Algorithms," 1998.

[3] A. Bazma, I. JONASSEN, I. EIDHAMMER, and D. GILBERT, "Approaches to the Automatic Discovery of Patterns in Biosequences," *Journal of Computational Biology*, vol. 5, pp. 279–305, Jan. 1998.

[4] A. S. Park and J. R. Glass, "Unsupervised Pattern Discovery in Speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 186–197, Jan. 2008.

[5] Y. Zhang and J. R. Glass, "Towards multi-speaker unsupervised speech pattern discovery," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4366–4369, IEEE, 2010.

[6] S. Salvador and P. Chan, "FastDTW : Toward Accurate Dynamic Time Warping in Linear Time and Space,"

[7] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," pp. 159–165, May 1990.

[8] L. Rabiner, A. Rosenberg, and S. Levinson, "Considerations in dynamic time warping algorithms for discrete word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 575–582, Dec. 1978.

[9] Y. Xie and B. Wiltgen, "Adaptive Feature Based Dynamic Time Warping," vol. 10, no. 1, pp. 264–273, 2010.

[10] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*, (New York, New York, USA), pp. 1033–1040, ACM Press, June 2006.

[11] A. W.-C. Fu, E. Keogh, L. Y. H. Lau, C. A. Ratanamahatana, and R. C.-W. Wong, "Scaling and time warping in time series querying," *The VLDB Journal*, vol. 17, pp. 899–921, Mar. 2007.

[12] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, pp. 67–72, Feb. 1975.

[13] P. J. A. Cadzow, B. Baseghi, and T. Hsu, "Singular-value decomposition approach to time series modelling," vol. 130, no. 3, 1983.

[14] R. G. L. Doddington and George, "TIDIGITS Readme File," 1993.

[15] F. Mörchen, *Time series knowledge mining*. 2006.

[16] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, pp. 358–386, May 2004.

[17] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings 18th International Conference on Data Engineering*, pp. 673–684, IEEE Comput. Soc, 2002.

[18] D. J. Marchette, "Local dimensionality reduction," vol. 14, ch. Dimensiona, p. 469, 1999.

[19] F. Korn, H. V. Jagadish, and C. Faloutsos, "Efficiently supporting ad hoc queries in large datasets of time sequences," in *Proceedings of the 1997 ACM SIGMOD international conference on Management of data - SIGMOD '97*, vol. 26, (New York, New York, USA), pp. 289–300, ACM Press, June 1997.

[20] E. K. Chotirat Ann Ratanamahatana, "Three Myths about Dynamic Time Warping Data," *Mining, in the Proceedings of SIAM International Conference on Data Mining*, 2005.

[21] M. De Wachter, M. Matton, K. Demuynck, P. Wambacq, R. Cools, and D. Van Compernolle, "Template-Based Continuous Speech Recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1377–1390, May 2007.

[22] M. De Wachter, K. Demuynck, P. Wambacq, and D. Van Compernolle, "A locally weighted distance measure for example based speech recognition," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. I–181–4, IEEE, 2004.

[23] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid Evaluation of Speech Representations for Spoken Term Discovery,"

[24] E. Keogh and M. Pazzani, "Derivative dynamic time warping," *the 1st SIAM Int. Conf. on Data Mining (SDM- . . .*, 2001.

[25] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.

[26] L. Chun-Lin, "Chapter 1 Overview," ch. wavelets I, 2010.

[27] S. Mallet, *A Wavelet Tour of Signal Processing*. Academic Press ,London, 1998.

[28] H. Zhang, "Unsupervised Feature Extraction for Time Series Clustering Using Orthogonal Wavelet Transform.," *Informatica (03505596)*, vol. 30, no. 3, pp. 305–319, 2006.

[29] K.-P. Chan and A.-C. Fu, "Efficient time series matching by wavelets," in *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, pp. 126–133, IEEE, 1999.

[30] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, pp. 909–996, Oct. 1988.

[31] J. Gorman, O. Mitchell, and F. Kuhl, "Partial shape recognition using dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 257–266, Mar. 1988.

[32] M. A. Haider, *Morphometric analysis of temporally varying shapes in 3 dimensions*. Bsc honors project, University of Edinburgh, 2012.

[33] C. D. Boor, *A practical guide to splines*. 1978.

[34] S. Osowski and D. D. Nghia, "Fourier and wavelet descriptors for shape recognition using neural networksa comparative study," *Pattern Recognition*, vol. 35, pp. 1949–1957, Sept. 2002.

[35] R. Bracewell, *The Fourier transform and its applications*. 1986.

[36] L. . R. Keogh, E., Zhu, Q., Hu, B., Hao. Y., Xi, X., Wei, "UCR Time Series Data sets," 2011.