DATA PREPROCESSING

THE AUTHOR

The Dynamic Time Warping(DTW) algorithm is the one of the oldest algorithms that is used to compare and cluster sequences varying in time, length and speed. Formally, given two temporal sequences, the algorithm utilises the technique of dynamic programming to find an optimum alignment between through the computation of local distances between the points in each sequence. The time and computational complexity of this algorithm is $O(\text{mn})$ where m and n denote the length of the sequences that are being compared. Thus for high dimensional time series sequences, the time and computational costs incurred by the algorithm are quite high which makes DTW a very unattractive choice for clustering or discovering motifs in high dimensional data sets. Intuitively speaking, DTW is a clustering algorithm that clusters similar patterns varying in time and speed. Another drawback for working in high-dimensional spaces is the contrast between the distances of nearest and furthest points. The distances between such points become increasingly smaller as the dimensionality increases. This makes it difficult to construct meaningful cluster groups in such spaces.

To address the issue of the curse of dimensionality, DTW algorithms employ a window constraint to reduce the search space. The window constraints determine the allowable shapes that a warping path can take. To reduce the time and computational costs incurred by the algorithm, the window size is reduced as the dimensionality of the data increases. Rigid constraints impose a more rigid alignment that prevents an overly temporal skew between two sequences, by keeping frames of one sequence from getting too far from the other. For clustering data sets such speech utterances, the effect produced by such global constraints is highly undesirable. If we consider two utterances of a word spoken at different time frames, the patterns 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. Thus it is necessary to explore techniques other than window constraints that can improve the performance of the DTW algorithm in terms of both accuracy and time.

Before investigating methods to improve a technique, it is highly necessary to first understand the nature of the data itself. In this chapter, I investigate data-driven preprocessing techniques that attempt to understand the underlying intrinsic structure of the lower-dimensional space on which the data lives. By achieving a thorough understanding of the data,we can achieve dimensionality reduction by isolating and identifying smaller set of new(current) features that are more relevant for the problem in hand.

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. One of the most popular approaches to feature selection 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 maximize our natural pattern-recognition abilities, such as using multiple plots per page.

Feature extraction processes on the other hand are concerned with a 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 variable. Hence 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 as shown by figure 1:

In the rest of this chapter, I explore a range of feature selection and extraction methods and investigate whether their application can improve the performance of the DTW algorithm in terms of both accuracy and time complexity.

1. Feature Selection

The computational and time complexity associated with the DTW algorithm is governed by the dimensionality of the time series. To get a feel of the data, I employed exploratory

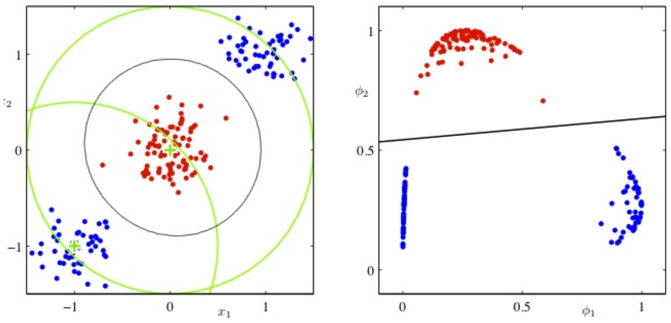


FIGURE 1. The figure on the right corresponds to location of the data points in the feature space spanned by gaussian basis functions $\phi_1(x)$ and $\phi_2(x)$

data analysis on the isolated word utterances belonging to the test and training data sets that I constructed from the TIDIGITS corpus. The aim here to identify and isolate redundant features from the time series data. To get an idea about the structure of the data, I have studied the plots of the time series sequences along with performing auditory perception on the individual samples. Figure 2 gives 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. These durations of silence segments are considerably long compared to the interesting regions in the acoustic signal that actually contain information about the spoken digit . Removing these silence segments not only reduce the dimensionality of the time series but also results in minimal loss of information.

- Through auditory perception of numerous samples, I have discovered that the recordings are highly distorted when played in matlab even when the data is scaled so that the sound is played as loud as possible without clipping. The distorted signal fails to provide any time auditory clue about category of the speaker i.e whether the speaker belongs to { boy,girl, men,women} and the signal must be played multiple times for its class to be correctly identified.

From further experiments, I have seen that if I down-sample the utterances by $\frac{1}{2}$ which in other words means decreasing the sampling frequency by half, the resultant sampled signal is much clearer to understand. Sub-sampling the utterances by half involves removing every other sample from the time series. From the observation of figure 3, it can be sen that this technique keeps the global trend of the signal intact but results in the loss of local information. Furthermore through auditory perception of the sampled signals, I have

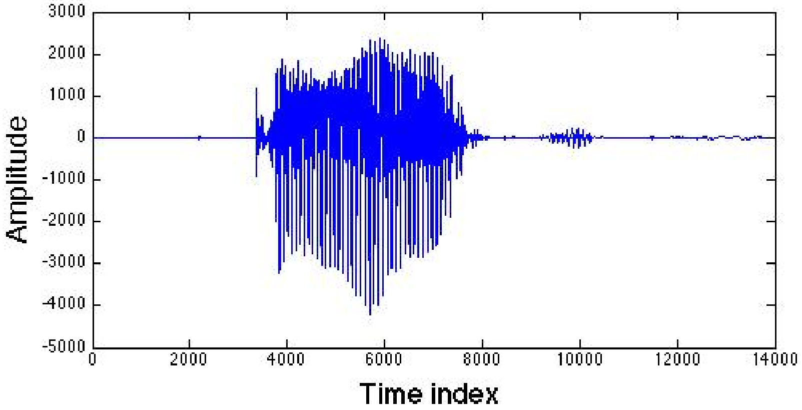


Figure 2. ‘Raw ’signal

discovered that losing some **local information** actually cleans the signal in a manner that allows the listener to identity the speaker’s category and the utterance’s class with ease.

With this knowledge I have constructed the following algorithm: ‘signal filter‘ that achieves feature selection by removing segments of silence and downsampling the remaining segments by half. An outline of the algorithm is as follows:

Algorithm 1 SignalFilter

```
procedure SignalFilter(signal)                                    ▷ raw signal
    $threshold = 0$
    maxAmplitude= max(rawSignal)
    Adapt the threshold based on the value taken by the maximum amplitude
    signalSil_R← removeSilence(rawSignal,threshold)
    return output← downsample signalSil_R by $\frac{1}{2}$
end procedure
```

- The algorithm removes all samples in the times series sequence whose magnitude is less than the threshold. The threshold used is an adaptive parameter. By using the information of the signal’s maximum amplitude the algorithm sets the threshold accordingly. It raises the threshold for signals corresponding to speakers having a loud and deep voice and lowers the threshold for signals corresponding to speakers having gentle and low voice.

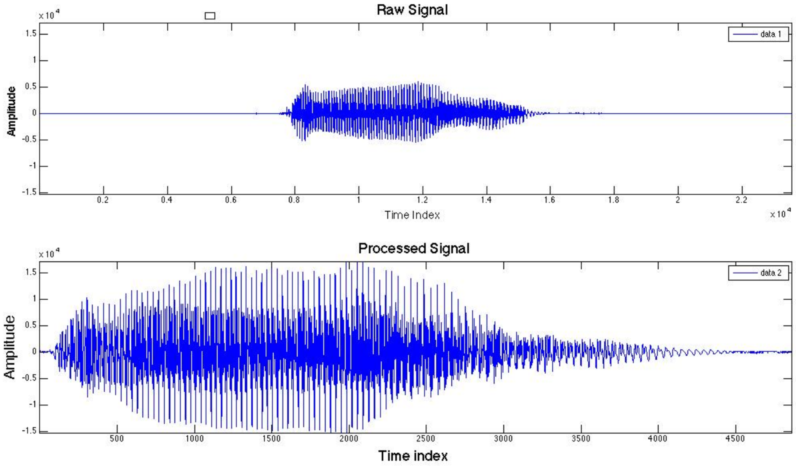


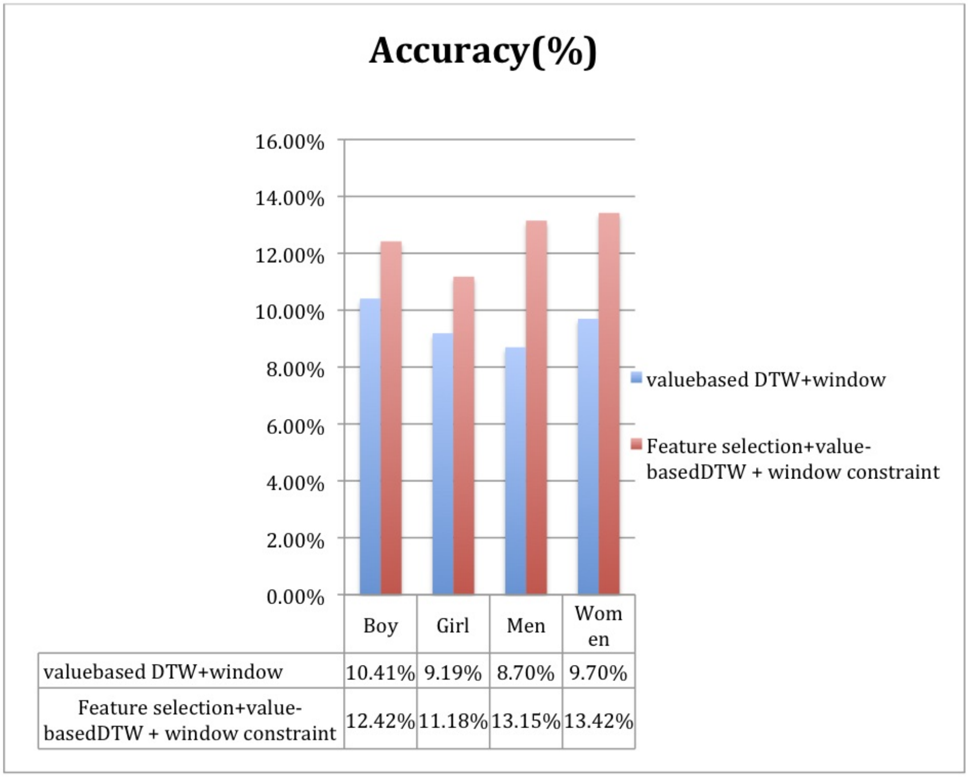
FIGURE 3. shows the raw acoustic signal corresponding to the utterances of the digit '5' alongside with the version that has its dimensionality reduced by the filter discussed above. From the comparison of the plots, it can be observed that the filter preserves the interesting patterns associated with the utterance while succeeding in reducing the dimensionality of the data.

To analyse how introducing this feature selection process effects the performance of a DTW algorithm in working with high dimensional data, I have run the baseline value-added DTW(i.e DTW using raw values) twice: once using the feature selection process as a preprocessing step and once without the feature selection process. An outline of the algorithm is given below.

Algorithm 2 Value-Based DTW

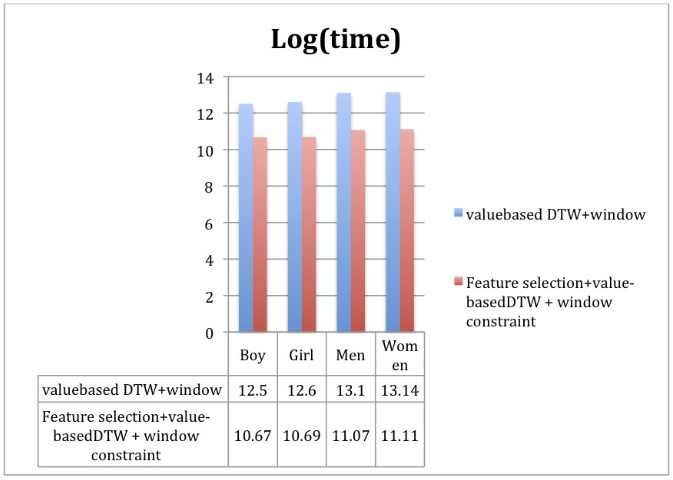
```
procedure VALUE-BASED($seq1, seq2$)                                ▷ two raw sequences
    w = max($\lceil 0.1 * max(n.m) \rceil$, abs(n-m))                  ▷ Window constraint
    for i=1: to length(seq1) do                        ▷ Initialise the DTW cost matrix
        DTW(i,0) = $\infty$
    end for
    for i=1 to length(seq2) do
        DTW(0,i) = $\infty$
    end for
    for i=2 to length(seq1) do
        for j=max(2, i-w) to min(length(seq2), i+w) do          ▷ cost(a,b)≡euclid(a,b)
            DTW(i,j) = cost(seq1(i),seq2(j)) + min{ DTW(i-1,j)+DTW(i,j-1)+DTW(i-1,j-1)}
        end for
    end for
    return result = $\frac{\text{DTW(n,m))}}{nm}$                   ▷ n=length(seq1), m=length(seq2)
end procedure
```

The focus of my research here 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 the dimensionality of the time series sequences is still very high. Thus for these experiments I have employed the most rigid window constraint w = max($\lceil 0.1 * max(n.m) \rceil$, abs(n-m)) that keeps frames from one sequence from getting too far from the other. The results found from the experiments are as follows:



	Boy	Girl	Men	Wom en
valuebased DTW+window	10.41%	9.19%	8.70%	9.70%
Feature selection+value-basedDTW + window constraint	12.42%	11.18%	13.15%	13.42%

Figure 4



	Boy	Girl	Men	Wom en
valuebased DTW+window	12.5	12.6	13.1	13.14
Feature selection+value-basedDTW + window constraint	10.67	10.69	11.07	11.11

Figure 5

Observations:

- Employing the prior feature selection process allows DTW subjected to global window constraint to improve both its accuracy and time complexity.

  Explanations:

  - The DTW algorithm,due to the high dimensionality of the time series sequences is subjected to a window constraint that forces the algorithm to operate on the diagonal region of the DTW cost matrix. Removing these redundant features increases the accuracy because these features primarily represent segments of silence and since all utterances contain silence segments, taking these silences into account degrades the performance as they bring i an unwanted notion of similarity in dissimilar patterns.
  - The size of the DTW cost matrix is $O(\text{mn})$. Achieving dimensionality reduction through feature selection reduces the size of the cost matrix and thus decreases the computational cost.

2. Feature extraction

To improve the performance of the DTW algorithm even further, in this section I investigate domain-independent and domain dependent feature extraction methodologies that employ an appropriate functional mapping to extract features that capture the intrinsic patterns of the data. The motivation behind this approach is to investigate to what degree we can improve the performance of the standard algorithm across different domains without making changes to the algorithm itself.

2.1. **Domain-independent feature extraction.** The fundamental problem of baseline (value-based) DTW is that the numerical value of a data point in a 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 know as *Derivative* DTW is proposed but it 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). Ideally we need to use features that contains information about the overall shapes of the sequences plus the local trend around the points. This allows the DTW to built a complete picture of the data point in relation to the rest of the sequence and hence achieve a better optimal alignment between the two sequences.

For feature extraction, the methodology that I have used for this setup is based on Xie and Wiltgen's paper[]. In their paper, the authors highlight a domain-independent feature extraction process where each point in the time series sequence is replaced by a 4 dimensional vector. In this vector, the first two features correspond to information regarding the local trends around a point and the last two features reflect the position of that point in the global shape of the sequence. From the experiments conducted on the UCR data sets, it has been observed that embedding DTW with this feature extraction process yields greater accuracy across all datasets.

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

$$f_{\text{local}}(r_i) = (r_i - r_{i-1}, r_i - r_{i+1})$$

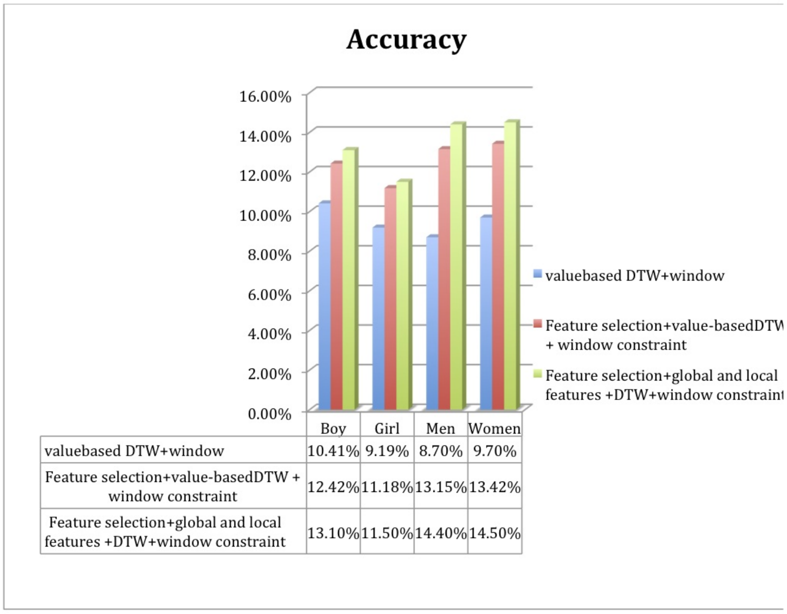
The extraction of global features is constrained by two factors: the features that reflect information about global trends and the features must be in the same scaling order as the local features. Being in the same scale allows them to be combined with local features. In [] the authors used the following method to extract global features from the time series sequence:

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

Note : The local and global features have no definition for the first and last points in a sequence.

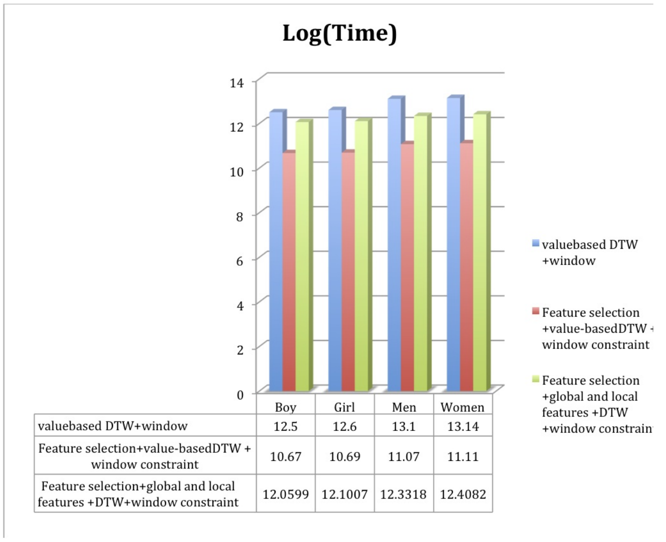
When working with high dimensional time series data, the main drawback of employing this feature extraction method is that it does not offer the advantage of dimensionality reduction. The dimensionality of the feature space is just two dimensions less than the dimensionality of the original data. The DTW algorithm combined with this feature extraction process therefore suffers from the curse of dimensionality as before. To tackle this issue, as a prior step to the feature extraction process, I applied the feature selection process that I have discussed in the previous section to remove redundant features.

The length of the resultant sequence of the vectors is still large. To tackle the problem of high computational cost, I have used the same window constraint applied the DTW algorithm equipped the two preprocessing stages of feature selection and extraction on the test data set that I had constructed from the TIGITS corpus. A summary of the results are given below:



	Boy	Girl	Men	Women
valuebased DTW+window	10.41%	9.19%	8.70%	9.70%
Feature selection+value-basedDTW + window constraint	12.42%	11.18%	13.15%	13.42%
Feature selection+global and local features +DTW+window constraint	13.10%	11.50%	14.40%	14.50%

FIGURE 6



	Boy	Girl	Men	Women
valuebased DTW+window	12.5	12.6	13.1	13.14
Feature selection+value-basedDTW + window constraint	10.67	10.69	11.07	11.11
Feature selection+global and local features +DTW+window constraint	12.0599	12.1007	12.3318	12.4082

FIGURE 7

2.2. **Domain-dependent feature extraction.** Speech is usually segmented in frames of 20 to 30 ms, and the window analysis is shifted by 10 ms. Each frame is converted to 12 MFCCs Less correlated than spectral features easier to model than spectral features MFCCs are not robust against noise

the goal is to transform the input wave transform into a sequence of feature vectors. For speech data the most commonly used is the MFCC-mel cepstrum ceptral 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 th e key attribute in providing useful information about phones. Cepstrum provides a useful way to separate the information of the vocal tract from the glottal source.

1) boosts the energy of the signal at high frequencies to improve phone detection 2) partitions the time series sequence into frames using a hamming window 3) to extract spectral information at different frequency bands 4) transforming to mel scale improves recognition performance as it models the property of human hearing 5) makes the feature less sensitive to variations in input such as power variations on the speakers mouth. 6) take the first 12 cepstral values from spectrum of the log of the spectrum