BRIEF ARTICLE

THE AUTHOR

1. Report

In the previous chapter, we have seen that using a MFCC representation of utterances with regions of silence removed leads to a large improvement in accuracy, time and computational complexity in the performance of DTW algorithm augmented with a euclidean metric..The main contributing factor behind the large time and computational complexity of the base line DTW is the **size** of the time series sequences. The computational cost of a DTW algorithm is (mn) where m and n denote the length of the two time series sequences currently compared. Using longer sequences increases the size of the DTW cost matrix hence resulting into a greater number of computations.

The DTW algorithm on its own is a domain independent algorithm that uses a similarity metric to compare any two sequences through comparison of their global trends. The algorithm employes dynamic programming to search a space of mapping between the time axis of the two respective sequences to determine the optimum alignment between them. The only difference between MFCC-augmented DTW and baseline DTW is the feature extraction stage. In machine learning, feature extraction refers to the pre-processing stage that involves the extraction of new features from a set of raw attributes through a suitable functional mapping. The extraction phase of MFCC features involves a segmentation of the time series followed by a functional mapping on the segmented windows. The resultant sequence of extracted feature vectors has a much smaller length compared to the length of the original sequence. Evident from the experiments done in the previous chapters, the use mel-cepstrum features extracted on 'cleaned' signals not only increases the accuracy of DTW but also reduces the time and computational cost through reduction of dimensionality of the original sequence.

Any time series sequence contains both local and global trends. In some time series datasets[], it has been observed that incorporating the information about these local trends and global shapes in the clustering /classification process does improve the performance of the DTW. The feature extraction methodologies used for their works are domain and application independent. The MFCC feature extraction on the other hand, is a domain and application dependent. This feature extraction process can only be applied to time series

sequences corresponding to speech. From a scientific stand point, it will be interesting to compare and see how well/bad the domain independent methods are to domain dependent methods. In this chapter, I investigate an unsupervised methodology that:

- incorporates information about local and global trends in the feature extraction process
- employs an adaptive DTW that tackles the issue of the large time and computational complexity by moving from working on time series sequences to sequences of segmented time-slices. To counter the tradeoff in the decrease in accuracy, the algorithm is equipped with a kernel function(self-proposed) that is designed to measure the similarity of sub-sequences more accurately than standard euclidean metric by being invariant toward time-dilation and scale.

2. Feature extraction

The fundamental problem of baseline 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 the fundamental problem with this DTW is that it fails to detect significant 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[]. Each point in the time series sequence is replaced by a 4 dimensional vector where 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.

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:

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

3. Adaption of DTW

The feature extraction methodology discussed above maps the time series sequence to a a time series sequence of vectors whose length is $\|X_n\| - 2$. ($\|X_n\|$ denotes the original length of the time series sequence). The DTW augmented with these features will still suffer from large time and computational complexity as before. In the MFCC feature extraction, the time series sequence is first segmented into series of frames of length 20ms. Through appropriate functional mapping, the frames are then converted into vectors. Because the length of the resultant sequence of vectors is much smaller than the length of the original time series, the size of the DTW cost matrix is therefore smaller than before hence decreasing the time and computational cost associated with each comparison.

Using this as motivation, the time series of 4d vectors extracted in the previous stage are segmented using windows of width 5ms. The number of segmented windows is much smaller than the length of sequences. Thus if we adapt the DTW to work on sequence of frames rather than the time series sequences, we can achieve a large improvement in the time and computational cost associated in the testing phase.

The problem now can be shifted to finding an appropriate kernel that can be used to compute the similarity between two subsequences of feature vectors while adjusting to the temporal skewness present within the subsequences. Intuitively speaking , we need a kernel that behaves like a DTW on pairs of subsequences.

The motivation behind the kernel proposed for this problem comes from the polynomial kernel. Lets consider the following example. Let x and z be two dimensional vectors. Consider the simple polynomial kernel of degree 2 :$k(x, z) = (x^T z)^2$. This kernel can expressed as :

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

Hence the 2nd order polynomial kernel is equivalent to a corresponding feature mapping ϕ that contains terms of order 2. Now, if we generalise this notion then $k(x, z) = (x^T z)^M$ contains all monomials of order M. For instance, if x and z are two images, then the kernel represents a particular weighted sum of all possible products of M pixels in the first image with M pixels in the second image.

Now consider the following signals: The signal denoted by the 'blue' color is a 'slower'

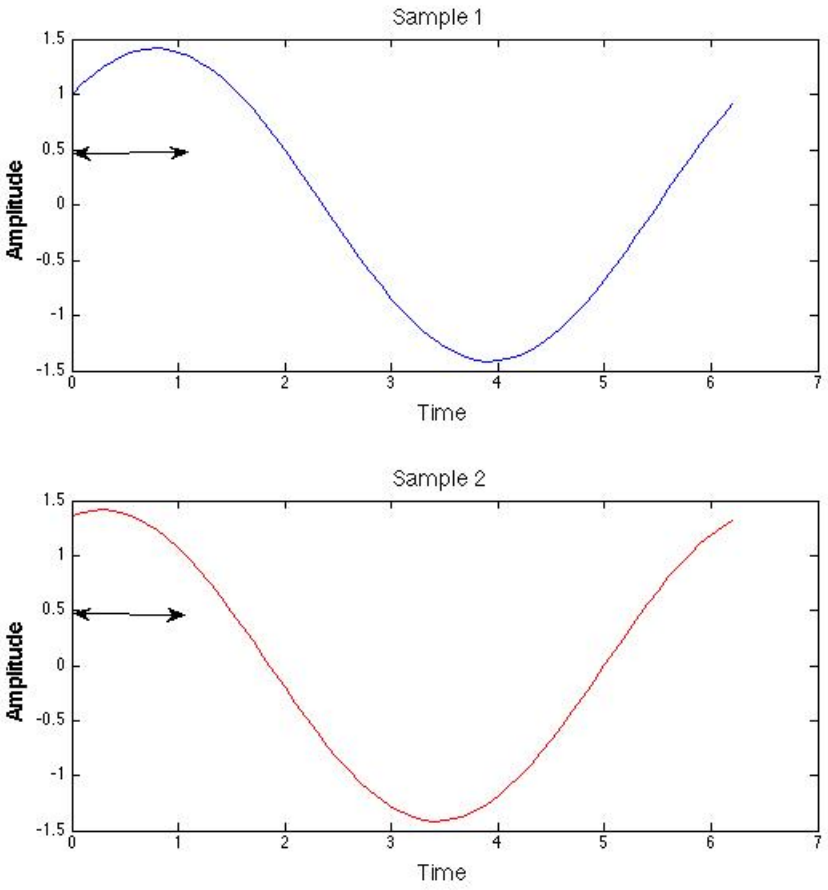


FIGURE 1. Two signals separated by translation

version of the signal denoted by the 'red' color . In the above example, lets consider the window spanned by the arrows:

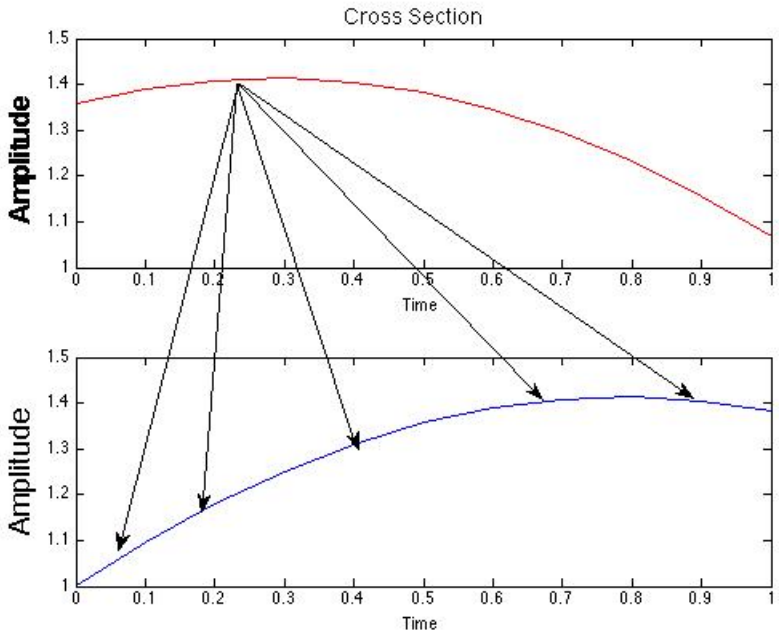


FIGURE 2. Two identical subsequences varying in time

Ideally, we want a metric that takes into account the variation of speed and time when comparing two similar subsequences. We will want to compare the global and local properties associated with a point in one subsequence with the global and local properties of points at different regions in the second sub-sequence illustrated by figure 2. Using a euclidean metric in this scenario is inappropriate. The euclidean metric in this context is identical to linear time warping where the two subsequences will be matched based on a linear match of the two temporal dimensions. In our context, we need a kernel that computes the similarity between two sub-sequences by warping the time axis. From the earlier discussion of the polynomial kernel, we have seen that if we consider the generalised case then $k(x, z) = (x^T z)^M$ represents a particular weighted sum of all possible products of M time series points in the first sub-sequence with M time series points in the second subsequence here x and z represent respective subsequences. Using this as motivation I propose the following kernel:.

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

where n denotes the length of the window and x_i and z_j represents the 4-dimensional features indexed by the points in two sub-sequences.

For a windows of size 3, this kernel metric representation can be expanded and expressed as :

$$\begin{aligned} k(x,z) &= <\sum_{i=1}^{3} x_i, \sum_{j=1}^{3} z_i> \\ &= <(x_1+x_2+x_3),(z_1+z_2+z_3)> \\ &= <x_1,z_1>+<x_1,z_2>+<x_1,z_2>+<x_2,z_1>+<x_2,z_2>+<x_2,z_3>+..... \end{aligned}$$

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 $\{(x_i z_i)|x_i \in \text{seq1}, z_i \in \text{seq2}\}$. Similar to the polynomial kernel, the proposed kernel allows us to match all possible pairs of vectors belonging to the tow sub-sequences.

It is easy to check that this proposed kernel is in fact a valid kernel:

- K(x,z)= K(z,x) ⇒the function is symmetric.
- The kernel satisfies Mercer's theorem : K(x,z) $=\phi(x)^T\phi(x)$ where the feature mapping corresponds to $\phi(y) = \sum_{i=1}^{3} y_i$.

Kernels constructed from dot products measure the similarity between two samples. For this kernel to be used as an appropriate cost function in the DTW algorithm, we need:

(1) the codomain to be in the range from 0 to ∞.

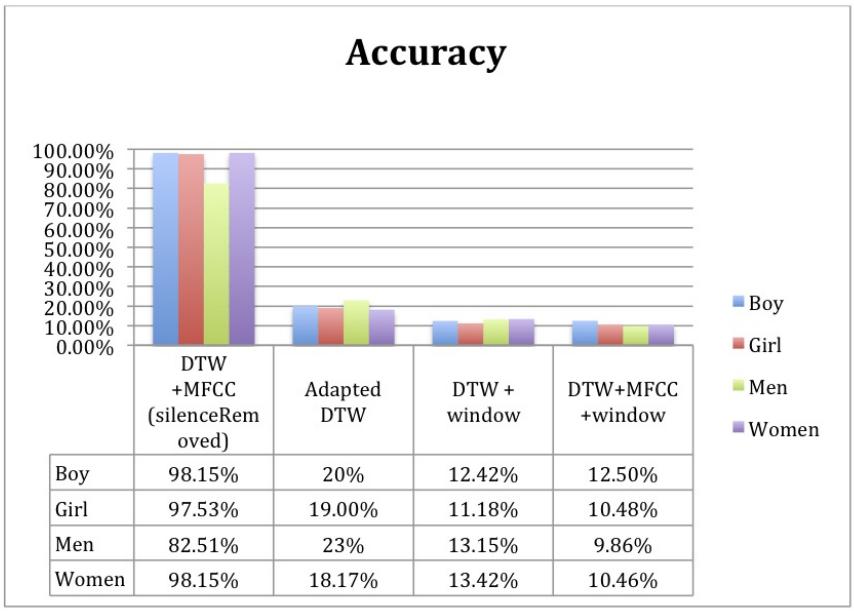
(2) Larger values given by the kernel to signify great degree of dissimilarity and smaller values to signify a high degree of similitude.

An ideal cost function that make use of dot product is the *arc-cosine*. Hence for my experiments, I augmented the following cost function to be DTW algorithm:

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

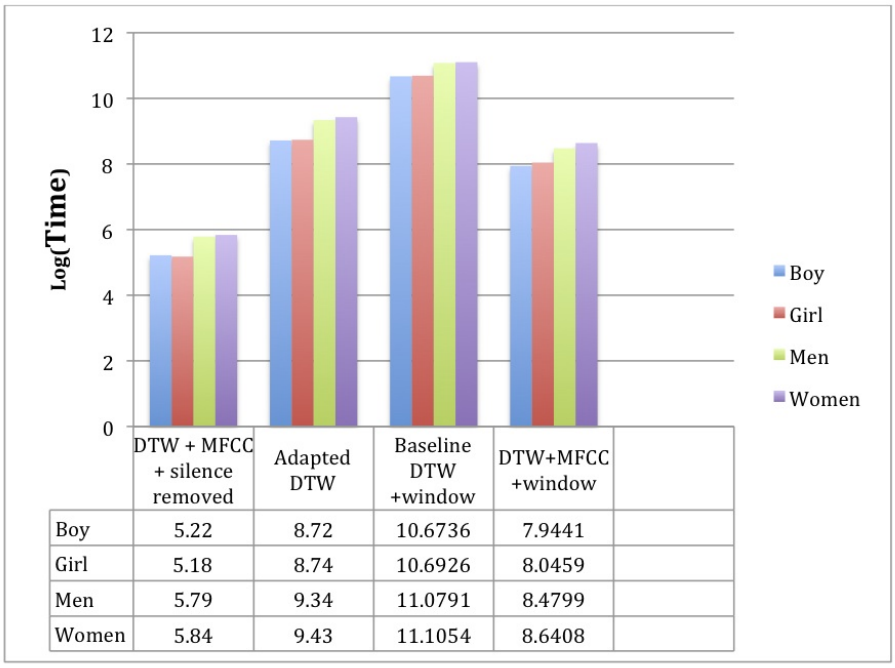
where $X = \sum_{i=1}^{n} x_i$ and $Z = \sum_{j=1}^{n} z_i$

4. Experimental results



	DTW +MFCC (silenceRemoved)	Adapted DTW	DTW + window	DTW+MFCC +window
Boy	98.15%	20%	12.42%	12.50%
Girl	97.53%	19.00%	11.18%	10.48%
Men	82.51%	23%	13.15%	9.86%
Women	98.15%	18.17%	13.42%	10.46%

FIGURE 3. Accuracy



	DTW + MFCC + silence removed	Adapted DTW	Baseline DTW +window	DTW+MFCC +window	
Boy	5.22	8.72	10.6736	7.9441	
Girl	5.18	8.74	10.6926	8.0459	
Men	5.79	9.34	11.0791	8.4799	
Women	5.84	9.43	11.1054	8.6408	

FIGURE 4. Time complexity