Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping

Nicholas Gillian
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
ngillian01@qub.ac.uk

R. Benjamin Knapp
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
b.knapp@qub.ac.uk

Sile O'Modhrain
Sonic Arts Research Centre
Queen's University Belfast
United Kingdom
sile@qub.ac.uk

ABSTRACT

This paper presents a novel algorithm that has been specifically designed for the recognition of multivariate temporal musical gestures. The algorithm is based on Dynamic Time Warping and has been extended to classify any N-dimensional signal, automatically compute a classification threshold to reject any data that is not a valid gesture and be quickly trained with a low number of training examples. The algorithm is evaluated using a database of 10 temporal gestures performed by 10 participants achieving an average cross-validation result of 99%.

Keywords

Dynamic Time Warping, Gesture Recognition, Musician-Computer Interaction, Multivariate Temporal Gestures

1. INTRODUCTION

Musicians commonly use body movements such as hand, arm and head gestures to communicate with other performers live on stage. This method of interaction is still difficult, however, between a musician and a computer despite the accessibility of cheap sensor devices and flexible machine learning software that can be used to recognise such gestures. Musical gestures can be difficult for a computer to recognise because many gestures are not simply static postures but consist of a cohesive sequence of movements that occur over a variable time period. Further, these temporal gestures commonly require multiple sensors to adequately capture the movement and a computer must therefore construct a model that describes not only the relationship between all the sensors at time t, but also how this relationship changes over time. Training a computer to automatically recognise *musical* temporal gestures also creates a number of interesting challenges that are not commonly found in other areas of human-computer interaction (HCI). This is because a musician will frequently want to use their own sensor technology to capture gestures that are inherently personal to that one performer; using the recognition of these gestures to interact with a specific piece of real-time audio performance software. The algorithms used to recognise a performer's gestures cannot therefore, in many instances, be pre-trained prior to being distributed to a musician; but must instead be trained by the musician. A musician therefore requires a recognition algorithm that can be quickly trained with a few examples of the performer's gestures, captured by whatever sensor is most applicable for that performer. The recognition algorithm employed to classify such gestures should, therefore, not be constrained to only recognise the gestures captured by one specific sensor, such as an accelerometer or webcam, but should work with any N-dimensional temporal signal. The key concept about designing and evaluating such an algorithm for the recognition of musical gestures is that, unlike many other areas of machine learning, the goal of the algorithm should be to achieve a low intra-personal generalisation error for the one user that trained the algorithm as opposed to a low inter-personal generalisation error. This paper presents an algorithm that has been specifically designed for the recognition of temporal musical gestures. The algorithm is based on *Dynamic Time Warping* (DTW) and has been extended to classify any N-dimensional, also known as multivariate, signal, automatically compute a classification threshold to reject any data that is not a valid gesture and be quickly trained with a low number of training examples.

2. DYNAMIC TIME WARPING

Dynamic Time Warping is an algorithm that can compute the similarity between two time-series, even if the lengths of the time-series do not match. One of the main issues with using a distance measure (such as Euclidean distance) to measure the similarity between two time-series is that the results can sometimes be very unintuitive. If for example, two time-series are identical, but slightly out of phase with each other, then a distance measure such as the Euclidean distance will give a very poor similarity measure. Figure 1 illustrates this problem. DTW overcomes this limitation by ignoring both local and global shifts in the time dimension [13].

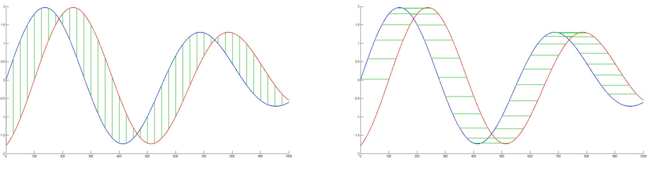
(a) Euclidean Distance (b) Dynamic Time Warping

Figure 1: Two identical time-series, slightly out of phase with each other, matched using Euclidean distance and Dynamic Time Warping

2.1 Related Work

There has been much work over the last two decades in applying DTW to such varying fields as database indexing

[6] [1], handwriting recognition [17] and gesture recognition [2] [4]. The vast majority of the recent work into DTW has focused on making the algorithm more computationally efficient [6] [7], with the time series in these works all being uni-dimensional signals. Proposed improvements to DTW included constraining the warping path [12] [5], lower-bounding [8] [10], numerosity reduction [19] and recursive resolution projection [13]. It has only been in recent years that research has been conducted into extending DTW to multiple dimensions, with the exception of the early work by Stettiner [14] who proposed an extension of DTW to multiple dimensions for the application of speech recognition. Vlachos et. al. [16] extended DTW to match two-dimensional time series. In previous work by Holt et. al. [15] and also separately by Ko et. al. [9], multi-dimensional DTW was achieved by using a distance function such as the absolute sum, Euclidean distance or cosine correlation coefficient to compute the distance over all the dimensions in the test time series with a template time series for each sample in time. The result of this distance function was used by the standard DTW algorithm to compute the warping cost between the test time series and the template time series. Wollmer et. al. [18] proposed a different approach to multi-dimensional DTW, using a three-dimensional distance matrix to compute the minimum distance between the input time series and a reference time series. This work used a bimodal input signal (speech data and gesture data captured by a mouse) and would therefore be computationally expensive to expand to an N-dimensional input stream as a large dimensional space would need to be constructed and navigated for each of the G gestures in the database.

Merrill et al. [11] successfully applied DTW to the recognition of musical gestures. Using the custom-built FlexiGesture (a two handed device that featured a number of sensors including accelerometers, gyroscopes, along with squeezing, bending and twisting sensors), a user could train the system to recognise up to 10 temporal gestures by pressing a 'trigger' button which started the data recording process, releasing the button when the gesture was completed. The system then asked the user to continually re-perform the gesture as it trained a template model for that gesture. Tests showed that the system was able to classify novel gestures into one of 10 classes with up to 98% accuracy.

2.2 One-Dimensional DTW

The foundation algorithm for DTW is as follows. Given two, one-dimensional, time-series, $\mathbf{x} = \{x_1, x_2, ..., x_{|\mathbf{x}|}\}^\intercal$ and $\mathbf{y} = \{y_1, y_2, ..., y_{|\mathbf{y}|}\}^\intercal$, with respective lengths $|\mathbf{x}|$ and $|\mathbf{y}|$, construct a *warping path* $\mathbf{w} = \{w_1, w_2, ..., w_{|\mathbf{w}|}\}^\intercal$ so that $|\mathbf{w}|$, the length of $\mathbf{w}$ is:

$$\max\{|\mathbf{x}|, |\mathbf{y}|\} \leq |\mathbf{w}| < |\mathbf{x}| + |\mathbf{y}| \tag{1}$$

where the kth value of $\mathbf{w}$ is given by:

$$\mathbf{w}_k = (\mathbf{x}_i, \mathbf{y}_j) \tag{2}$$

A number of constraints are placed on the warping path, which are as follows:

- The warping path must start at: $\mathbf{w}_1 = (1, 1)$
- The warping path must end at: $\mathbf{w}_{|\mathbf{w}|} = (|\mathbf{x}|, |\mathbf{y}|)$
- The warping path must be continuous, i.e. if $\mathbf{w}_k = (i, j)$ then $\mathbf{w}_{k+1}$ must equal either (i, j), $(i + 1, j)$, $(i, j + 1)$ or $(i + 1, j + 1)$
- The warping path must exhibit monotonic behavior, i.e. the warping path can not move backwards

There are exponentially many warping paths that satisfy the above conditions. However, we are only interested in finding the warping path that minimizes the normalised total warping cost given by:

$$\min \quad \frac{1}{|\mathbf{w}|} \sum_{k=1}^{|\mathbf{w}|} DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j}) \tag{3}$$

where $DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j})$ is the distance function (typically Euclidean) between point i in time-series $\mathbf{x}$ and point j in time-series $\mathbf{y}$, given by $\mathbf{w}_k$. The minimum total warping path can be found by using dynamic programming to fill a two-dimensional ($|\mathbf{x}|$ by $|\mathbf{y}|$) cost matrix $\mathbf{C}$. Each cell in the cost matrix represents the accumulated minimum warping cost so far in the warping between the time-series $\mathbf{x}$ and $\mathbf{y}$ up to the position of that cell. The value in the cell at $\mathbf{C}_{(i,j)}$ is therefore given by:

$$\mathbf{C}_{(i,j)} = DIST(i, j) + \min\{\mathbf{C}_{(i-1,j)}, \mathbf{C}_{(i,j-1)}, \mathbf{C}_{(i-1,j-1)}\} \tag{4}$$

which is the distance between point i in the time-series $\mathbf{x}$ and point j in the time-series $\mathbf{y}$, plus the minimum accumulated distance from the three previous cells that neighbor the cell i,j (the cell above it, the cell to its left and the cell at its diagonal). When the cost matrix has been filled, the minimum possible warping path can easily be calculated by navigating through the cost matrix in reverse order, starting at $\mathbf{C}_{(|\mathbf{x}|,|\mathbf{y}|)}$, until cell $\mathbf{C}_{(1,1)}$ has been reached, as illustrated in Figure 2. At each step, the cells to the left, above and diagonally of the current cell are searched to find the minimum value. The cell with the minimum value is then moved to and the previous three cell search is repeated until $\mathbf{C}_{(1,1)}$ has been reached. The warping path then gives the minimum normalised total warping distance between $\mathbf{x}$ and $\mathbf{y}$:

$$DTW(\mathbf{x}, \mathbf{y}) = \frac{1}{|\mathbf{w}|} \sum_{k=1}^{|\mathbf{w}|} DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j}) \tag{5}$$

Here, $\frac{1}{|\mathbf{w}|}$ is used as a normalisation factor to allow the comparison of warping paths of varying lengths.

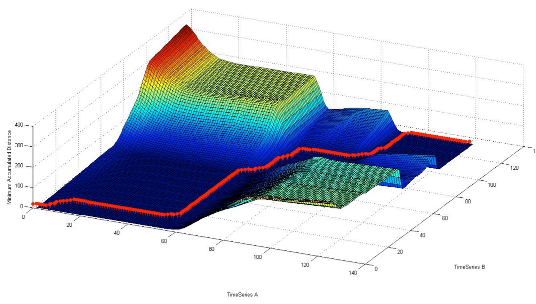

Figure 2: Cost Matrix and the Minimum Warp Path through it (indicated by the red line)

2.3 Numerosity Reduction

DTW is a useful tool for computing the distance between two time-series. It is, however, a computational costly algorithm to use for real-time recognition, as every value in the cost matrix must be filled. Clearly this is unusable for real-time recognition purposes, particularly if the unknown time-series is being matched against a large database of gestures. To speed up both the training of the gesture templates and the real-time classification of an unknown N-dimensional input time-series, we tested various methods of numerosity reduction. Perhaps one of the most rudimentary methods for numerosity reduction is to downsample the time-series by a factor of n. To avoid aliasing, the data is filtered using a low-pass FIR filter with a rectangular window and a filter order of n.

2.4 Constraining the Warping Path

Another method commonly adopted for improving the efficiency of DTW is to constrain the warping path so that the maximum warping path allowed cannot drift too far from the diagonal. Controlling the size of this warping window will greatly affect the speed of the DTW computation. If the warping window is small, a large proportion of the cost matrix does not need to be searched or even constructed. The size of the warping window can be controlled by varying the parameter r, given as the percentage of the length of the template time-series. The warping window is then set as the distance, r, from the diagonal to directly above and to the right of the diagonal. This type of global constraint is referred to as the Sakoe-Chiba band [12]. Itakura has also proposed another global constrained based on a parallelogram [5].

3. ND-DTW

Section 2.1 describes the standard implementation of DTW for two, uni-dimensional time-series. It is common, however, in computational fields such as gesture recognition to have time-series that feature multiple-dimensions, such as data captured by a 3-axis accelerometer. It is in this instance that we require an implementation of DTW that can compute the distance between two N-dimensional time-series. We will use the common approach used by [15][9] to compute the distance between two N-dimensional time-series. This takes the summation of distance errors between each dimension of an N-dimensional template and the new N-dimensional time-series. The total distance across all N dimensions is then used to construct the warping matrix $\mathbf{C}$. We will use the Euclidean distance as a distance measure across the N dimensions of the template and new time-series.

$$DIST(i,j) = \sqrt{\sum_{n=1}^{N} (i_n - j_n)^2} \tag{6}$$

The following section describes our N-Dimensional Dynamic Time Warping (**ND-DTW**) algorithm. In the training stage, an N-dimensional template (ϕ_g) and threshold value (τ_g) for each of the G gestures is computed. In the real-time prediction stage a new N-dimensional time-series is classified against the template that gives the minimum normalised total warping distance between the N-dimensional template and the unknown N-dimensional time-series. We will now discuss each element of the algorithm in detail.

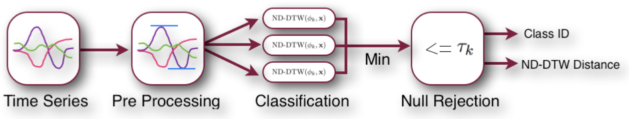

Figure 3: The ND-DTW classification chain

3.1 Training the ND-DTW Algorithm

In order for ND-DTW to be used as a real-time recognition algorithm, a template must first be created for each gesture that needs to be classified. A template can be computed by recording M_g training examples for each of the G gestures that are required to be recognised. After the training data has been recorded, each of the G templates can be found by computing the distance between each of the M_g training examples for the gth gesture and searching for the training example that provides the minimum normalised total warping distance when matched against the other M_g-1 training examples in that class. The gth template (ϕ_g) is therefore given by:

$$\phi_g = \arg\min_i \quad \frac{1}{M_g - 1} \sum_{j=1}^{M_g} \mathbf{1}\{\text{ND-DTW}(\mathbf{X}_i, \mathbf{X}_j)\} \qquad 1 \le i \le M_g \tag{7}$$

where the $\mathbf{1}\{\cdot\}$ that surrounds the ND-DTW function is the indicator bracket, giving 1 when $i \neq j$ or 0 otherwise and $\mathbf{X}_i$ and $\mathbf{X}_j$ are the ith and jth N-dimensional training examples for the gth gesture in the form of $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and $\mathbf{x} = \{x_1, x_2, ..., x_{|\mathbf{x}|}\}^\intercal$. The ND-DTW function in (7) is simply the extension of the standard DTW algorithm to N-dimensions:

$$\text{ND-DTW}(\mathbf{X}, \mathbf{Y}) = \quad \min \quad \frac{1}{|\mathbf{w}|} \sum_{k=1}^{|\mathbf{w}|} DIST(\mathbf{w}_{k_i}, \mathbf{w}_{k_j})$$

$$DIST(i,j) = \sqrt{\sum_{n=1}^{N} (i_n - j_n)^2} \tag{8}$$

3.2 Multi-Threaded Training

One major advantage of using the DTW algorithm is that each template (i.e. each gesture) can be computed independently from the other templates. This is of particular use on new machines that feature multiple processors as a multi-threaded training approach can be adopted in which each template's training routine is launched in a separate thread. This training approach greatly speeds up the overall training time for a DTW classification system as one template does not need to wait for the previous template to be trained before it can start its own training routine.

The DTW algorithm also has one other advantage in that, if a new gesture is added to an existing trained model or an existing gesture is removed, the entire model does not need to be retrained. Instead, a new template and threshold value only needs to be trained for the new gesture, thus greatly reducing the training time. If an existing gesture is removed from the model then no re-training is required as the DTW classification system simply removes this template and threshold value from its 'database'. This is not the case for other machine learning algorithms, such as an Artificial Neural Network, as the entire system would need to be retrained from scratch any time a new gesture is added or removed.

3.3 Classification Using ND-DTW

After the ND-DTW algorithm has been trained, an unknown N-dimensional time-series $\mathbf{X}$ can be classified by computing the normalised total warping distance between $\mathbf{X}$ and each of the G templates in the model. c, the classification index representing the gth gesture is then given by finding the corresponding template that gave the minimum normalised total warping distance:

$$c = \arg\min_g \quad \text{ND-DTW}(\phi_g, \mathbf{X}) \qquad 1 \le g \le G \tag{9}$$

3.4 Determining the Classification Threshold

Using equation (9) $\mathbf{X}$, an unknown N-dimensional time-series, can be classified by calculating the distance between it and all the templates in the model. The unknown time-series $\mathbf{X}$ can then be classified against the template that results in the lowest normalised total warping distance. This method will, however, give false positives if the N-dimensional input time-series $\mathbf{X}$ is in-fact not made up of any of the gestures in the model. This false classification problem can be mitigated by determining a classification threshold for each template gesture during the training phase. In the prediction phase, a gesture will only be classified against the template that results in the lowest normalised total warping distance, if this distance is less than or equal to the gesture's classification threshold. If the distance is above the classification threshold, then the algorithm will classify the gesture against a null class, indicating that no match was found:

$$\hat{c} = \begin{cases} c & if(d \leq \tau_g) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where c is given by equation (9), d is the total normalised warping distance between ϕ_g and $\mathbf{X}$ and τ_g is the classification threshold for the gth template.

The classification threshold for each template can be set as the average total normalised warping distance between ϕ_g and the other $M_g - 1$ training examples for that gesture, plus γ standard deviations:

$$\tau_g = \mu_g + (\sigma_g \gamma) \tag{11}$$

where

$$\mu_g = \frac{1}{M_g - 1} \sum_{i=1}^{M_g} \mathbf{1}\{\text{ND-DTW}(\phi_g, \mathbf{X}_i)\} \tag{12}$$

$$\sigma_g = \sqrt{\frac{1}{M_g - 2} \sum_{i=1}^{M_g} \mathbf{1}\{(\text{ND-DTW}(\phi_g, \mathbf{X}_i) - \mu_g)^2\}} \tag{13}$$

where the $\mathbf{1}\{\cdot\}$ that surrounds the ND-DTW function is the indicator bracket, giving 1 when $i \neq$ the index of the training example that gave the minimum normalised total warping distance when matched against the other M_g-1 training examples in that class (i.e. the template) or 0 otherwise and $\mathbf{X}_i$ is the ith training example for the gth class. γ can be initially set to a number of standard deviations (e.g. 2) during the training phase and later adjusted by the user in the real-time predication phase until a suitable classification/rejection level has been achieved.

It is critical when calculating the classification threshold for each of the g gestures to perform any preprocessing such as scaling or downsampling in the same order as it would be performed during the real-time classification stage. If this is not completed in the same order then the optimal classification threshold will not be found. We will now discuss the various preprocessing options that can be used for ND-DTW.

3.5 Preprocessing for ND-DTW

Pre-processing is necessary for ND-DTW if either (a) any of the N-dimensional data originate from a different source range or (b) if invariance to spatial variability and variability of signal magnitude is desired. We now discuss both of these points and give appropriate preprocessing solutions for each.

3.5.1 *Varying Input Source Ranges*

It is important for each of the N-dimensional data in the time-series $\mathbf{X}$ to originate from a common source range. If this is not the case then one or more of the dimensions may heavily weight the results of the DTW. If each of the N-dimensional data do not originate from a common source range then each channel should be scaled using min-max normalisation prior to both the training of the templates and real-time prediction.

3.5.2 *Invariance to Amplitude & Spatial Variability*

Spatial variance and variability in the signal amplitude can be mitigated by first z-normalising both the input time-series and also the recognition templates. Z-normalisation will give both the input and template time-series zero mean and unit variance, therefore removing any affect that spatial variation or variability in the signal amplitude may have had. Keogh et. al. [7] also proposed using the derivative of the input signals to account for similar spatial problems. This method was also used successfully by Holt et. al. [15].

3.6 Real-time Implementation

The ND-DTW algorithm has been fully integrated into the SEC[1], a machine learning toolbox that has been specifically developed for musician-computer interaction [3]. The SEC is a third party toolbox consisting of a large number of machine learning algorithms that have been added to EyesWeb[2], a free open software platform that was established to support the development of real-time multimodal distributed interactive applications.

4. DTW EXPERIMENTS

Three experiments were run to validate the classification abilities of the ND-DTW algorithm. To test the algorithm 10 participants were recruited and asked to perform 25 repetitions of 10 gestures. The 10 gestures consisted of 'air drawing' several numbers and shapes with the right hand, including the numbers 1 -5, a square, a circle, a triangle, a horizontal line similar to a downbeat conducting gesture and a vertical line similar to a sidebeat conducting gesture. Each participant wore a Polhemus magnetic tracking sensor mounted on their right wrist which was sampled at 120Hz. The data collected from all 10 participants will be referred to as the numbers-shapes dataset. Because the ND-DTW algorithm has been specifically designed for the recognition of musical gestures, with the objective of creating an algorithm that can be quickly trained to accurately classify the musical gestures of the one performer that trained it, each experiment will validate the intra-personal generalisation abilities of the algorithm as opposed to the inter-personal generalisation.

4.0.1 *Experiment A*

This experiment tests the ND-DTW algorithm's ability to correctly classify the pre-segmented data from the numbers-shapes dataset. For each participant, a ND-DTW model was trained using 10-fold cross-validation, with the average cross-validation ratio (**ACVR**) taken over all 10 participants being used to evaluate the algorithm. This experiment was run with four conditions (C1) scaling off, z-normalisation off; (C2) scaling on, z-normalisation off; (C3) scaling off, z-normalisation on and (C4) scaling on z-normalisation on. γ was set to 2 and a downsample factor of 5 was used for all conditions. Condition C2 achieved the

[1]http://www.somasa.qub.ac.uk/ ngillian/SEC.html
[2]http://musart.dist.unige.it/EywMain.html

maximum ACVR of 99.37%, however the other conditions also achieved excellent classification results of 98.85% for C1, 98.95% for C3 and 99.37% for C4. This test shows that the ND-DTW algorithm provides excellent classification results on pre-segmented data. The ND-DTW algorithm achieved a perfect recognition result of 100% for several participants, with the algorithm achieving a classification result of over 99% for all but 1 participant. Figure 4 shows the cross-validation results for each of the 10 participants.

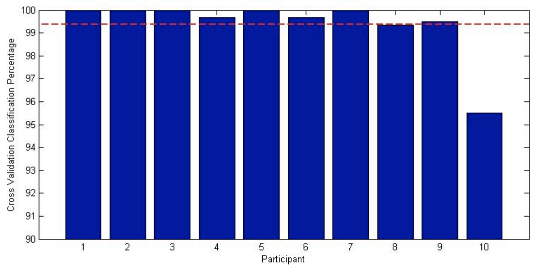

Figure 4: The cross-validation classification results for each of the 10 participants in condition C2. The ACVR is illustrated by the dotted horizontal line

4.0.2 *Experiment B*

This experiment tests the classification abilities of the ND-DTW algorithm with respect to a minimal amount of training data. This is an important test for music as, if a model can achieve as good a classification result with 2 training examples as it can with 20 training examples, then a performer can save time in both collecting the training data and also in training the model. For each participant, a ND-DTW model was trained using η randomly selected training examples from each of the 10 gestures and tested with the remaining data. η ranged from 3 - 20, starting at 3 as opposed to 1 because at least 3 training examples are required to estimate the threshold value for each template and stopping at 20 to allow at least 5 test examples per trial. To ensure that the results of this test were not weighted by a 'lucky' random selection of the best template from the 25 training samples of each gesture, each test for η was repeated 10 times and the average correct classification ratio (**ACCR**) was recorded and used for validation of the algorithm. γ was set to 2 for this experiment and a downsample factor of 5 was used. Figure 5 shows the ACCR for each iteration of η. This test shows that the number of training examples significantly effects the classification abilities of the ND-DTW algorithm. The ND-DTW algorithm achieved a moderate ACCR value of 74.74% with just 3 training examples. With 20 training examples it was able to achieve an ACCR value of 92.19%. It should be noted that the standard deviation over each iteration of η and across all 10 participants was very high. This shows that the classification abilities of the ND-DTW algorithm is heavily dependent on getting 'the best' training examples. Several participants, for example, achieved an ACCR value of $> 90\%$ with just 3 training examples. The same participants, however, also achieved an ACCR value of $< 70\%$ with the same number of training examples, showing that the 'quality' of the training examples heavily influences the results of the classification algorithm. The results of this test suggest that at least 11 training examples are required per-gesture if the user wants to achieve a robust classification result of $> 90\%$.

4.0.3 *Experiment C*

This experiment tests the ND-DTW algorithm's ability to correctly classify data from the numbers-shapes dataset in

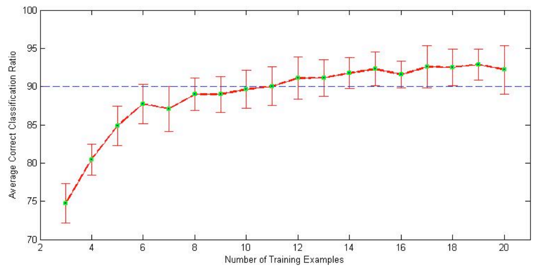

Figure 5: The ACCR values averaged across all 10 participants for each iteration of η. The horizontal blue line indicates the minimal training examples required to achieve a classification result of $> 90\%$

a continuous stream of data that also contains a number of null gestures. This evaluates two important aspects of the ND-DTW algorithm for the recognition of multivariate temporal gestures. Namely the algorithm's ability to correctly classify a set of temporal gestures from a continuous stream of data and also the algorithm's ability to reject any null gesture that is not contained in the model's database.

For each participant, a ND-DTW model was trained using 12 randomly selected training examples from each of the 10 gestures. After each model had been trained it was tested using a continuous stream of data. The continuous stream of data originated from the data-collection phase of the numbers-gestures database and contains all of the participant's trial recordings. The continuous stream therefore contains not only all of the 25 gestures the participant performed (12 of which were used to train the model) but also, importantly, the participant's movements in between each trial along with the periods of rest.

The continuous stream was tested by running a sliding window of size w over the data stream in increments of 10. The window size, w, was individually calculated for each participant by taking the average length of the 10 ND-DTW templates for that participant. For the majority of the participants, w was 304, with the shortest window length of 248 and the longest window length of 368. At each increment, the data within the window was given to the ND- DTW model for classification. Each sample of data had been labelled with an ID tag (0 for a null-gesture or the gth class ID for an actual gesture). This ID tag was used to evaluate if the ND-DTW model had made the correct classification for each window of data. As some windows covered a section of data that contained half a gesture and noise, the classification results of a window were only counted if the maximum ID count within the window was greater than 80% of the length of the window. This test was evaluated using the average correct classification ratio (**ACCR**) given by the total number of counted correctly classified windows over the total number of counted windows. The average precision ratio (**APR**), average recall ratio (**ARR**) and average null recall ratio (**ANRR**) were also computed. These provided an indication of the exactness of the classifier for each gesture across all the participants ignoring the null gestures (APR), an indication of the performance of the classifier over a specific gesture across all participants ignoring the null gestures (ARR) and an indication of the performance of the classifier at correctly rejecting the null gestures (ANRR). γ was set to 5 and a downsample factor of 5 was used for this experiment.

This test was run with the same four conditions found in experiment A. The ACCR values for each of the four conditions were 83.31%, 84.18%, 74.15% and 74.15% respectively. Condition C2 with scaling on - z-normalisation off achieved the highest ACCR value of 84.18%. The max-

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
APR	0.91	0.89	0.80	0.79	0.92	0.95	0.83	0.96	0.95	0.96
ARR	0.93	0.78	0.85	0.81	0.82	0.66	0.94	0.93	0.92	0.90

Table 1: The average precision ratio (APR) and average recall ratio (ARR) for each gesture in condition C2

imum individual correct classification result of 95.23% was achieved by the algorithm for participant 1, while the algorithm achieved the minimum individual correct classification result of 64.09% for participant 8. Table 1 shows the APR and ARR results for condition C2, averaged over all 10 participants. The APR and ARR results show that the majority of classification errors were made by in the recall of the algorithm, as opposed to the precision of the algorithm. This shows that the ND-DTW algorithm made the majority of classification errors by misclassifying gesture i as a null gesture, rather than misclassifying gesture i as gesture j. The ANRR value of 0.88 indicates that the algorithm was successful at distinguishing a null-gesture from a gesture in the database 88% of the time.

These results suggest that the ND-DTW algorithm performed well at rejecting null gestures and also performed well at not misclassifying gesture i as gesture j. The main error that the ND-DTW algorithm made was in misclassifying gesture i as a null gesture. Increasing φ would have increased the threshold value for each gesture and therefore less gestures may have been misclassified as a null gesture. However, increasing this threshold value would have also increased the number of false-positive classifications (were a null gesture was falsely classified as gesture i). This problem illustrates the compromise that a user must make about the sensitivity of their classification system. Increasing the thresholding value will increase the likelihood that a gesture will be classified but it will also unfortunately increase the likelihood of false-positive misclassifications. It is for this specific reason that we have initially set the algorithm to calculate the threshold value as the mean plus two standard deviations of the error between the template and the remaining training examples for each gesture. The performer is then able to manually adjust this threshold value during the real-time 'live' prediction phase until the algorithm has reached a satisfactory recognition rate.

5. CONCLUSION

This paper has presented the ND-DTW algorithm which has been specifically designed for the recognition of multivariate temporal musical gestures. Three experiments have validated the algorithms ability to correctly classify a set of multivariate temporal gestures with a limited number of training examples and from a continuous stream of data that also contains null-gestures.

6. REFERENCES

[1] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[2] K. Forbes and E. Fiume. An efficient search algorithm for motion data using weighted pca. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, page 76. ACM, 2005.

[3] N. Gillian, R. B. Knapp, and S. O'Modhrain. A machine learning toolbox for musician computer interaction. In *NIME11*, 2011.

[4] A. Heloir, N. Courty, S. Gibet, and F. Multon. Temporal alignment of communicative gesture sequences. *Computer Animation and Virtual Worlds*, 17(3-4):347, 2006.

[5] F. Itakura. Minimum prediction residual principle applied to speech recognition. *Readings in speech recognition*, page 154, 1990.

[6] E. Keogh and M. Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289. ACM, 2000.

[7] E. Keogh and M. Pazzani. Derivative dynamic time warping. In *First SIAM international conference on data mining.* Citeseer, 2001.

[8] E. Keogh and C. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[9] M. Ko, G. West, S. Venkatesh, and M. Kumar. Using dynamic time warping for online temporal fusion in multisensor systems. *Information Fusion*, 9(3):370–388, 2008.

[10] D. Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition*, 42(9):2169–2180, 2009.

[11] D. J. Merrill and J. A. Paradiso. Personalization, expressivity, and learnability of an implicit mapping strategy for physical interfaces. *CHI2005*, 2005.

[12] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Readings in speech recognition*, page 159, 1990.

[13] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[14] Y. Stettiner, D. Malah, and D. Chazan. Dynamic time warping with path control and non-local cost. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 1994.

[15] G. ten Holt, M. Reinders, and E. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In *Thirteenth annual conference of the Advanced School for Computing and Imaging*, 2007.

[16] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. *Proceedings of the 9th ACM SIGKDD int. conf. on Knowledge discovery and data mining*, 2003.

[17] V. Vuori, J. Laaksonen, E. Oja, and J. Kangas. Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters. *International Journal on Document Analysis and Recognition*, 3:150–159, 2001.

[18] M. Wullmer, M. Al-Hames, F. Eyben, B. Schuller, and G. Rigoll. A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. *Neurocomputing*, 2009.

[19] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. Ratanamahatana. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*, page 1040. ACM, 2006.