ROBUST TIME-SERIES RETRIEVAL USING ADAPTIVE SEGMENTAL ALIGNMENT

by

SHAHRIAR SHARIAT TALKHOONCHE

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Vladimir Pavlovic

And approved by

____

____

____

____

New Brunswick, New Jersey

October, 2013

ABSTRACT OF THE DISSERTATION

Robust Time-Series Retrieval Using Adaptive Segmental Alignment

By Shahriar Shariat Talkhoonche

Dissertation Director: Vladimir Pavlovic

The problem of time-series retrieval arises in many fields of science and constitutes many important sub-problems including indexing, storage, representation, similarity measurement, etc. The center piece of time-series retrieval is, however, measurement of similarity between the query and the stored sequences in the data-base. Since different time-series sampled from similar phenomena can have variable lengths and/or warping, simple distance metrics such as Euclidean distance are either undefined or do not provide an accurate similarity measure. Therefore, alignment methods such as dynamic time warping have been proposed. They essentially rely on the distance between every sample point of contrasting sequences and recover their alignment using dynamic programming. These algorithms are effective when the sequences are noise-free and causal.

In this work we introduce the concept of segmental sequence alignment. We claim that dynamically dividing the contrasting sequences into subsequences and recovering the optimal and monotonic matching between them instead of individual time-points can result in constructing a similarity measure more robust to noise and non-causality. We propose two different approaches and variants of them to accomplish segmental sequence alignment.

The first proposed approach is an isotonic extension of Canonical Correlation Analysis (CCA) properly constrained to satisfy the time monotonicity constraint necessary for an alignment algorithm. The second approach is an extension of pair-HMM, which is a probabilistic model for aligning sequences. We have defined a proper observation model and efficient learning and inference algorithms to jointly recover the segmentation and alignment from segmental pair-HMM. We also propose a relaxation to the probabilistic model to increase the computational efficiency.

We have shown the utility of our proposed techniques through extensive experiments on both synthetic and real-world data. We have applied our methods to various data sets from EEG signals to human activity. Our methods showed generally significant improvement over traditional models especially in instances when the sequences are corrupted by high levels of noise or are locally non-causal.

Acknowledgements

This thesis would not have been possible without continuous guidance, and support from my advisor Prof. Vladimir Pavlovic throughout all these years of my PhD study.

I would like to thank all my lab-mates for attending my presentations and giving me helpful comments.

I would like to thank my committee members Prof. Casimir Kulikowski, Prof. Alexander Schliep and Prof. Fernando De La Torre. I had the privilege of being a student of Prof. Kulikowski in his course on machine learning and his TA for two semesters. His influence on my research career is very significant and I learned a lot about philosophy of machine learning and scientific presentation from him. The last course that I took in my PhD was Introduction to Bioinformatics taught by Prof. Schliep. I learned about pair-HMM in his class and the extension to that model constitutes half of my thesis. Nothing more is needed to be said about his contribution to my research! I was also Prof. Shliep's TA for the same course a year later. I always enjoyed Prof. De La Torre's talks and papers. His paper on Canonical Time Warping inspired me to invent the first model that I present in this thesis.

I was a TA throughout my PhD studies and this is a fact that I learned a lot from my students. Most important of all, they taught the value of patience. No one can undermine the exposure to such diverse set of minds. I had students from many countries with different level of enthusiasm and knowledge. Each one of them taught me how different and yet similar the human beings are.

Dedication

To my late father, without whose encouragement I would not have even started my studies in PhD level. To my dear wife without whom I had given up a long time ago. To my mother who gave up her desire of having her last child by her side so he can find his passion in a far away land.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

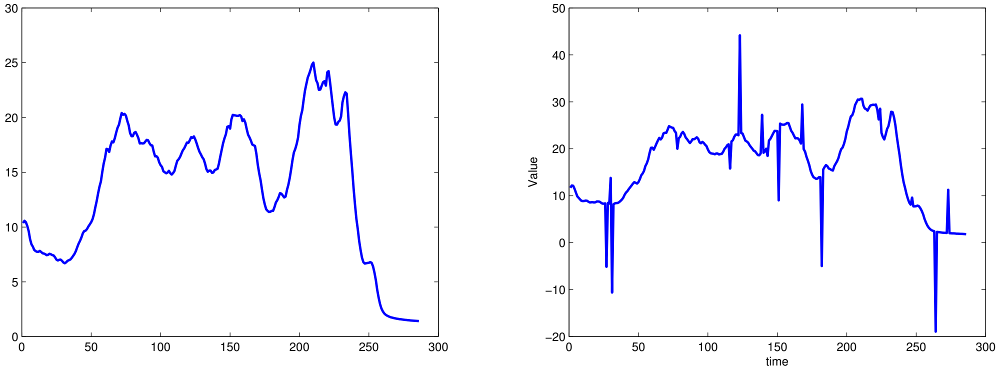
Rapid growth of information technology industry resulted in generating massive amounts of data. Mining through this mountain of data is a difficult yet essential task. The difficulty arises from the fact that even though the speed of an information retrieval system is an important factor but so is its correctness.

A vast amount of the generated data can be expressed in form of an ordered set of measurements. Examples of such representation range from stock closing prices ordered (indexed) by days to video sequences that are an ordered set of images. Biological sequences such as DNA or RNA are another examples. One can even represent the shape of an object by moving clock-wise around the object and sample the coordinates of every point.

This type of data formation is typically referred to as *time-series* or *time sequence*[1]. One can define the time-series retrieval as the task of *finding the most similar time-series to a query sequence in a database and returning its class label.* Many recognition tasks ranging from financial data analysis such as sector recognition [1, 2, 3] to human activity recognition [4, 5, 6] can be posed as a time-series retrieval problem. There have been many attempts to improve the accuracy and scalability of algorithms involved in time-series mining and retrieval [7, 8].

To achieve a reliable, fast and robust time-series retrieval, one may need to solve many subproblems such as: What is the appropriate representation? How should the data be indexed? How is the database serialized and stored? Finally, how does one

[1] We use *time-series*, *sequence* and, in some instances, *signal* interchangeably. In the last case we obviously refer to time signals.



(a) Original time-series

(b) Noisy time-series (impulse noise)

Figure 1.1: A sample of an original signal and its noisy version. This type of noise is called impulse or spike noise and happens due to sudden and temporary sensor failure or interference.

assert the similarity of the query time-series to another sequence stored in the database.

In this work we focus on the last question. We will design a similarity measure that has two specific requirements that are overlooked by other measures, resulting in sub-optimal retrieval and classification performance. Those two requirements are:

- Resilience to noise and
- Resilience to local non-causality.

Similarity measures for time-series are typically based on matching and comparing individual time samples. This approach makes similarity measures sensitive to noise. There are many real-world applications that might carry different amounts of noise. Essentially the amount of noise depends on the environment, sensors, data acquisition method, measurement error and the sampling method.

Stock prices, for example, are noise-free but sampling from and the motion of an object can be very noisy due to all the above reasons. If the data is passed through a noisy channel or the process naturally happens in an environment that contains many sources of noise, the amount of noise can be very significant. For instance, consider the electroencephalogram (EEG) signals. These signals are recorded by attaching a set of sensors to a cap and positioning it on subject's head. The sensors detect the voltage fluctuations caused by neural activity. These recordings often carry significant amount of noise caused by different sources. Scalp itself acts as a low-pass filter and sensors in local proximity impose magnetic interference on each other. Therefore, a simple noise removal pre-processing is not effective. A simple similarity measure in this case will easily fail to provide a meaningful measure of distance between sequences of EEG signals.

Sudden sensor failures are also among very common source of noise (Figure 1.1). A faulty sensor might record long noisy samples and the noise-free data might not be recoverable. This type of noise usually appears as random impulses over the time span of the signal. Some similarity measures remove this type of noise by thresholding the point-to-point distance. However, finding a suitable threshold is not an easy task and usually is accomplished through some pre-sets based on engineering knowledge and domain expert recommendations. Therefore, designing a similarity measure that is able to remove the noise adaptively and according to the properties of the dataset is highly desirable.

While the first requirement, i.e. resilience to noise, seems natural specially for real-world applications, the second one, i.e. resilience to local non-causality needs more clarification. First, let us define causality. A causal time-series has ordered time-points. This specifically means that the time-series has been received exactly in the same order that has been transmitted from the source.

For instance, if one samples from the closing price of the stock market every day and store the samples (without passing them through a possibly non-causal channel), the stored signal is considered causal. Now, for the sake of this particular example, assume that the samples are transmitted to a distant destination using radio signals which are relayed through many hops. It is clear that there exist several paths for a sample to reach the destination and different paths might have different delays. Now assume that the delay for the sample transmitted at time t is more than the one transmitted at time $t+1$. The different delays cause the samples to be received in the reverse order. Thus, the result of passing our causal signal through a channel with variable delays might be a shuffled (non-causal) signal at the destination (Figure 1.2).

Of course, in the case of radio signals and network packets one is probably able to

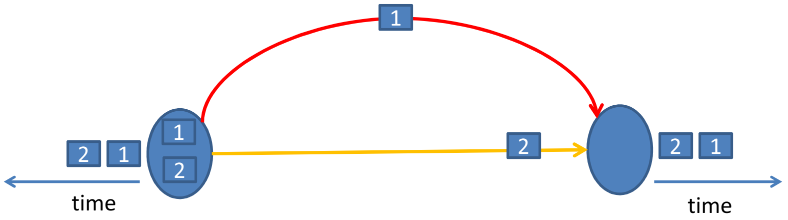


Figure 1.2: Different routes with different delays from source to destination cause an ordered set samples to be received in the reverse order.

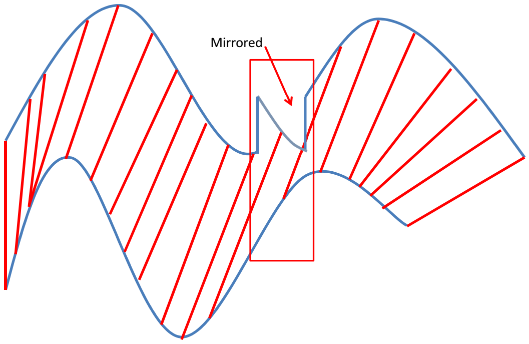


Figure 1.3: A small portion of the signal is mirrored, which might look like noise!

add a time-stamp to the data at the source and thereby recover the correctly-ordered signal in the destination. In many cases however, it is not possible to add anything to the signal because the source is not in our control. For instance , consider the brain signals [9] or cosmic rays [10].

It is also noteworthy that some irregularities in signals that are often counted as noise can easily be seen as a local shuffling of the true signal (Figure 1.3). Therefore, in those cases it is crucial for the similarity measure to be able to find the most relevant signal despite the local shuffling of the samples.

In Chapter 3 [11] we will discuss our motivation for designing a time-series alignment algorithm robust to noise and non-causality after first discussing prior approaches related to this general problem in Chapter 2.

1.2 Thesis Statement and Contributions

Statement and Hypothesis: In this thesis we claim that dynamically dividing the contrasting sequences into local subsequences and monotonically matching them instead of individual time-points can result in constructing a similarity measure more robust to noise and non-causality compared to the state-of-the-art alignment algorithms. We prove our hypothesis through theoretical results backed by extensive experimentation.

We essentially propose to divide the sequences into segments and relate the similarity of the contrasting time-series to the distance of these segments. The intuition behind segmental alignment is that comparing statistics of two small subsequences instead of two sample points must be more robust to noise. Also, within that subsequence the order of points can be ignored and thus the local non-causality will be handled naturally.

We summarize the contributions of this thesis, addressing the design of a similarity measure resilient to noise and local non-causality.

- **We present a similarity measure by solving a regression problem:** In Chapter 4, we introduce the concept of segmental alignment. We propose a least squares objective based on a certain formulation of CCA. We pose the problem as a regression and impose a set of constraints to impose convexity of the multipliers and apply time monotonicity on the segment level. The proposed method works by finding the distance of the convex hulls of segments. From another point of view, we dynamically realize the segments and then reshape them such that they are as similar as possible. From CCA point of view, we transfer the sequences into another space where they are as similar as possible and then find the Euclidean distance of the embedded sequences [6].

- **We design a probabilistic similarity measure based on Hidden Markov Models:** In Chapter 5, we introduce another approach based on Hidden Markov Models (HMM) to recover better segments for alignment and thus construct a better similarity measure. We use a modified version of average linkage distance and transform it to a metric to measure the distance between segments efficiently. We also propose an appropriate learning algorithm that satisfies the constraints of an alignment algorithm. Through extensive experimentation we prove that using our new measure can result in high classification accuracy in presence of excessive amount of noise and non-causality [12]. We also propose a relaxation to our model to increase the computational efficiency. Bounding techniques are finally used on a certain data representation to prune unnecessary computation and speedup the process [13].

In Chapter 2, we review the literature on alignment approaches. We conclude this dissertation by having an overview of the performance of different similarity measures in Chapter 6.

Chapter 2

Basic Methodology and Review of Prior Work

2.1 Introduction

In this Chapter we discuss different similarity measures for time-series and mostly focus on alignment methods. Generally there are two distinct research paths in time-series alignment literature:

1. Model matching

2. Time-point to time-point matching.

Our focus in this work is mainly the second scenario. We however, briefly review a few works in the literature that can be categorized as "model matching".

2.2 Model Matching

When the objective is to align a query sequence to "all" sequences in the training set at once or assess whether the query sequence is likely to be aligned to the whole training set, one might consider building a generative model based on the training set and then compute the likelihood of observing the query sequence given the model.

The works in this line of research are usually built upon a *profile* of the data. Probably the most famous alignment model in this category is the profile HMM [14], which is typically designed for symbolic sequences. To construct a profile HMM, one needs to determine the number of states of the HMM and then the transitions and middle states associated with delete and insert operations are defined based on the profile model. The model can be trained using a regular expectation maximization approach typically used to train hidden Markov models. This model does not provide a pairwise similarity measure between two sequences but established the similarity in the class level.

An Extension of the profile HMM to real-valued time-series has been studied in several works in the literature. In [15], the authors propose that all samples of the data are generated by non-uniform sampling from a latent trace combined with rescaling and noise. They essentially assume a latent space with twice the length of the time-series in the dataset and the sub-sampling and scaling is are determined by state sequence. The observation model is assumed to be sampled from a Gaussian distribution while the transition weights are multinomial. The model is trained using a standard expectation maximization. They have applied their method to Liquid Chromatography - Mass Spectrometry and speed signals and show that this model indeed aligns the time-series.

The alignment problem can be tackled indirectly by estimating mixture of densities generating the samples in time. A successful attempt in this direction has been made by Kim and Pavlovic in [16]. The authors propose to consider a class conditional for each sequence by modelling it as a naive Bayes. Then they form an objective which is the summation of negative log likelihoods of the class conditionals and minimize it. Hence, a mixture model is defined over the dataset which can later be used to estimate the class label of the query. This approach for modelling the classification and retrieval of time-series has an intrinsic alignment mechanism through the mixture model generation.

In [17], Akimoto and Suemetsu propose to model the the sequences using Gaussian processes [18]. The idea is to have two processes, one for modelling the shape and another one for time transformation. Through the time transformation the misalignment in time is handled and through the shape process the scaling problem is solved. This type of modelling (time transformation and shape modelling) is typical in alignment literature [4, 19], but the novelty of this work is in modelling such functions through a Gaussian process prior.They infer from the model using Markov Chain Monte Carlo [20]. They show in the results that their model is competitive to the state-of-the-art and can recover the true alignment in case of moderately noisy input.

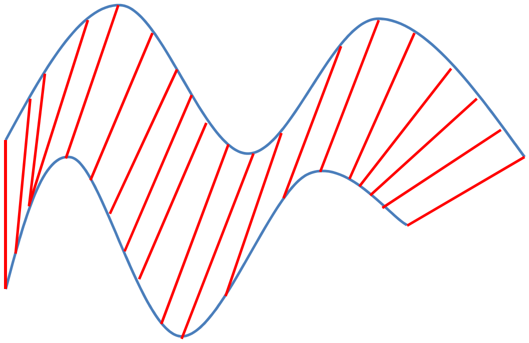


Figure 2.1: DTW is an instance of elastic similarity measures. It is in contrast to rigid measures such as Euclidean distance.

2.3 Time-Point Matching

Perhaps the most straightforward and intuitive measure of distance between two time-series is the Euclidean distance. The Euclidean distance is only defined when the contrasting sequences have the same length and dimension. Assume two time-series are given such that $x_i \in \mathbb{R}^d$ and $y_j \in \mathbb{R}^d$ and $X = \{x_i\}_{i=1}^N$ and $X = \{y_j\}_{j=1}^N$, the Euclidean distance is defined as

$$E(X, Y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2} \tag{2.1}$$

One can define other norms such as L_1 or L_∞ as well. The Euclidean distance has some advantages. First, its computation cost is linear in time length of the sequences. Second, it is parameter-free. Third, it is easy to implement and verify. In fact, in many cases the Euclidean distance is a competitive similarity measure when possibly combined with appropriate up-sampling and down-sampling to force the sequences to have the same length [21]. This distance however, is extremely sensitive to noise and misalignment and can deviate from the true and intuitive distance of sequences easily due to many artifacts that can be added to the time-series.

Wapring Problem: Euclidean distance assumes sample point at time (index) i in sequence X is associated with the ith sample point in sequence Y. Also it assumes that the contrasting sequences are of the same length. To relax these two very conservative assumptions, one can define a warping function, $w(\cdot)$, which deforms the time axis such that given this new indexing one sequence transforms to the contrasting one. In

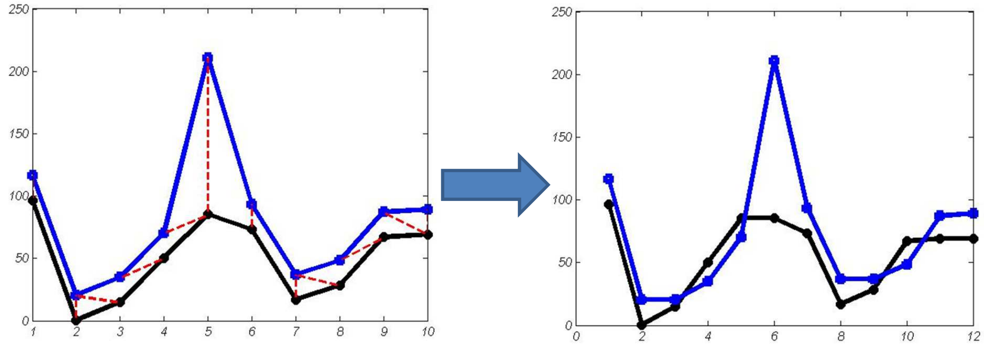


Figure 2.2: DTW attempts to stretch and displace the sequences so that they have the smallest possible distance. Left panel shows the original time-series and the correspondences recovered by DTW. Right panel illustrates the transformed time-series according to those correspondences.

particular a *warping problem* is defined as designing a warping function, $w(\cdot)$, such that $Y(t) = X(w(t))$ where t is the time or a non-decreasing index. The warping function needs to be monotonic and non-decreasing to preserve the direction of the time axis.

The rest of this Chapter is dedicated to reviewing different approaches that attack the warping problem from different angles. Some approaches try to estimate the warping function directly (for example [22]) and others try to focus on matching the time points and implicitly designing this function (for example [23]). The warping problem and the "global alignment problem" are essentially the same in the sense that both of them seek to find the most proper way of transforming one sequence to another.

The most famous alignment/warping approach or elastic similarity measure is Dynamic Time Warping (DTW) [24, 23]. DTW can be seen as the real-value version of the edit distance problem [25].This constitutes an *elastic* measure of distance (Figure 2.1). A dynamic programming algorithm is used to solve the edit distance problem that finds the minimum cost of transforming one sequence to the other one. As illustrated in Figure 2.2, DTW finds the minimum cost of transforming one sequence such that it is as similar as possible to the contrasting time-series. Since DTW is going to play a key role in this work we briefly review the algorithm and the constraints that are typically applied to it.

Dynamic Time Warping: Assume two time-series X and Y are given as before

Definition
$DTW(N,M) = \begin{cases} 0 & i=j=0 \\ \infty & i=0 \;\; or \;\; j=0 \\ D(x_i,y_i)+\min\{DTW(i-1,j-1), & i=1\dots N, \\ DTW(i-1,j), DTW(i,j-1)\} & j=1\dots M \end{cases}$
$ERP(N,M) = \begin{cases} \sum_1^N D(x_i,g), \sum_1^M D(g,y_j) & i=0 \;\; or \;\; j=0 \\ \min\{ERP(i-1,j-1)+D(x_i,y_j), & \\ ERP(i-1,j)+D(x_i,g), & i=1\dots N, \\ ERP(i,j-1)+D(g,y_j)\} & j=1\dots M \end{cases}$
$LCSS(N,M) = \begin{cases} 0 & i=0 \;\; or \;\; j=0 \\ & i=1\dots N \\ & j=1\dots M \\ LCSS(i-1,j-1)+1 & \|x_i-y_j\| \leq \epsilon \\ max\{LCSS(i-1,j), LCSS(i,j-1)\} & otherwise \end{cases}$
$EDR(N,M) = \begin{cases} N,M & i=0 \;\; or \;\; j=0 \\ \min\{EDR(i-1,j-1)+c, & i=1\dots N \\ EDR(i-1,j), EDR(i,j-1)\} & j=1\dots M \end{cases}$

Table 2.1: Dynamic programing based alignment algorithms. All algorithms are working on X and Y with N and M samples, respectively. $D(\cdot,\cdot)$ is a norm (usually L_1, L_2 or L_∞). $c=0$ is $|x_i-y_j| \leq \epsilon$ and c=1 otherwise.

but without the requirement of having the same length. That is $X = \{x_i\}_{i=1}^N$ and $Y = \{y_j\}_{j=1}^M$. The objective is to transform them so that they have the minimum possible distance. Assume that point to point distance is defined as the Euclidean distance. The dynamic programming is then defined as the first row of Table 2.1. Let us look at the dynamic programming recursion more carefully:

$$DTW(i,j) = D(x_i,y_i)+\min\{DTW(i-1,j-1), DTW(i-1,j), DTW(i,j-1)\}. \quad (2.2)$$

The matched points (i,j) (those that correspond to the minimum cost) constitute a path that is called alignment path or warping path or warping function (look at Figure 2.3). The constraints on DTW are:

1. **Boundary constraints**: The alignment must start from $(1,1)$ and end at (N,M).

2. **Continuity**: Warping path should not have any jumps. That is, every sample in one sequence must be associated with a sample from the contrasting sequence

3. **Time monotonicity**: The alignment path never goes back in time.

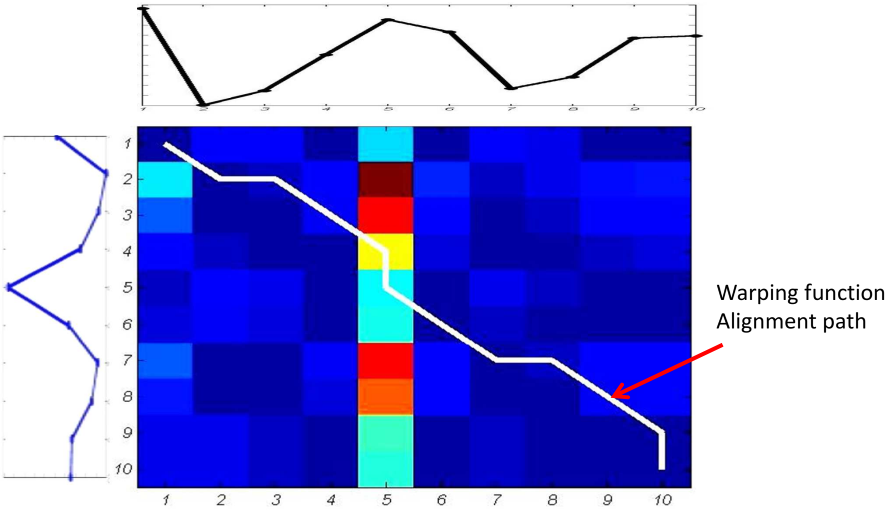
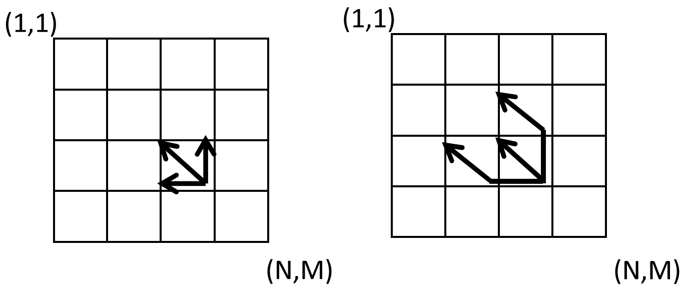


Figure 2.3: DTW alignment path for aligning two time-series. The alignment path is overplayed on the color coded pairwise distance of every points in contrasting sequences.

An important terminology is needed to be defined. Based on (2.2), the warping path can move in only three direction: horizontal (*deletion*), vertical (*insertion*) and diagonal (*match*) where all of them move forward in time to respect the third constraint, time monotonicity. Deletion and insertion can switch their meaning based on the reference sequence and they are called *gap* operations as well.

The time complexity of DTW is obviously $O(NMd)$. By keeping all the pairwise distances DTW can be solved in $O(NM)$. As a similarity measure one may take $DTW(N, M)$ as the distance of warped sequences and use it for retrieval. Despite the success in applying DTW for alignment and finding appropriate distance between time-series, there are cases where DTW fails to provide an intuitively correct alignment. In some cases, DTW might try to warp one axis to cover the variability in the other axis. That is, a large portion on one sequence might be associated with a single element in the contrasting sequence. This problem was originally detected by [24] and various methods were presented to handle it. One can categorize those attempts as the following:

Warping band: Perhaps the most famous method proposed by [24]. One can change (2.2) to $DTW(i, j) = \min\{\alpha DTW(i-1, j), \alpha DTW(i, j-1), DTW(i-1, j-1)\} + D(x_i, y_j)$ where α is positive real number. As α gets larger, the warping path will be more and more biased toward the diagonal. This idea can be implemented by



(a) Original single steps (b) A different step pattern

Figure 2.4: DTW with different step patterns. Panel (a) represents the classic DTW step pattern. Panel (b) shows the step pattern associated with: $DTW(i,j) = \min\{DTW(i-2,j-1), DTW(i-1,j-2), DTW(i-1,j-1)\} + D(x_i,y_j)$

hard coding the permissible off diagonal deviation proportional to the length of the sequences. We will elaborate more on this method in Chapter 4.

Windowing: Proposed by [23], one can restrict the elements of sequences that can be matched to those that fall into a window such that $|i - \left(\frac{N}{\frac{M}{j}}\right)| < R$, where R is a positive integer. One can consider other shapes instead of a square and such shapes have been proposed in several papers such as [26, 27].

Step patterns, (slope constraint): One can look at (2.2) as a set of permissible step patterns. For instance, one can change it to $DTW(i,j) = \min\{DTW(i-2,j-1), DTW(i-1,j-2), DTW(i-1,j-1)\} + D(x_i,y_j)$ which forces the warping path to move one step diagonally for each horizontal or vertical movement (Figure 2.4). Many stepping patterns have been proposed. [28] contains a review on those step functions.

In addition to the above methods, more grounded approaches such as Derivative DTW (DDTW) [29] have been proposed to alleviate the problems that arise when one applies DTW to sequences with local differences. Consider the case that X contains a time-point x_i and Y contains a time-point y_j of identical values but x_i is a part of a rising trend where y_j is part of falling trend. DTW considers matching x_i and y_j as a perfect match while intuitively this is not correct. In [29] Keogh and Pazzani propose to change the features that DTW works on to the derivatives of the sequences rather than their actual value. In fact, the feature they use is the average of the slope of the sequence at every point. The results show a better alignment can be achieved in certain cases by DDTW. Of course, one needs to deal with added noise resulting from derivative operation and therefore this method needs a noise removal pre-processing and if the data contains significant amount of noise, DDTW will not be effective at all.

One limitation of DTW (and many other alignment algorithms) is that they require the contrasting signals to have the same dimensionality. In other words, they do not allow aligning signals of different modalities. This is intuitively sensible since aligning two irrelevant signals (for example audio and video) does not make sense in the first glance. However, one can imagine numerous scenarios where one needs to find the relation of two signals of different modality. For instance, consider aligning audio recording and video capture of a speech given by a person. In this case one may want to overlay the two signals in the most appropriate way and thus needs an alignment tool.

In [4], the authors propose to accomplish the task of aligning signals of different modalities using a spatial embedding through Canonical Correlation Analysis (CCA) [30] and then aligning them in the common attribute space using DTW. They call this method Canonical Time Warping (CTW). The authors followed up with [22] by introducing Generalized Time Warping (GTW) to be able to align multiple sequences of different modalities efficiently by solving the objective function using Gauss-Newton algorithm and thus abandon the dynamic programming. There are other works in the literature of biological time-series alignment that are in the same line with CTW, such as [31] where the authors use CCA to find common attributes of multiple sequences. They parametrize the warping function by using a linear combination of hyperbolic tangents. They then call the sequences aligned in the attribute space. This approach results in finding the commonality of multiple sequences and the ability to isolate them from the irrelevant parts of very long sequences.

Dynamic Time warping, as we said before, is the real-value version of edit distance problem. Another category of alignment methods use a threshold on the distance of time samples to convert the problem back to the edit distance. That is, for every correspondence they compare the distance of contrasting samples against a threshold to decide whether they are similar or not. Many algorithms are designed based on this idea and they share many merits with string sequence alignment algorithms typically using in biological sequence alignment [14, 32]. The most famous algorithm in this category is Longest Common Sub-Sequence (LCSS) [33, 34]. In [34] Vlachos et al propose to use LCSS for retrieving trajectory sequences of at most three dimensions. They essentially propose to use the LCS algorithm by thresholding the distance of every two samples by ϵ and confine the warping window with another threshold δ. The similarity measure is then normalized by the length of the two sequences. They also show that this measure has a weak notion of triangular inequality and thus is suitable for retrieval. The thresholding enables LCSS to be more robust to noise and outliers compared to Euclidean distance and DTW.

Edit distance with Real Penalty (ERP) [35] is a distance metric which lives somewhere between DTW and LCSS. ERP does not compare the pairwise distance of sample points against a threshold. Instead, it compares the distance of each point to a reference in case of gap operations which can be interpreted as comparing the pairwise distance against a variable threshold. That is, if the distance between two points is too large it uses the distance of one of those points from the aforementioned reference point. Another work in this line is Edit Distance for Real Sequences (EDR) proposed in [36]. EDR advances the accuracy performance of LCSS by adding a constant gap penalty to the score function. The idea of gap penalty comes from biological sequence alignment [32]. In fact, there are three approaches one can take regarding gaps in an alignment algorithm: 1) no gap penalty (such as in edit distance or DTW); 2) Constant gap penalty as in EDR and 3) affine gap penalty. Look at [37] for a comprehensive review on different gap penalties. Generally, one can achieve better alignment results by using more sophisticated gap penalties such as affine. However, in that case there are more parameters to be determined. EDR achieves higher performance compared to DTW, ERP and LCSS in case of noisy data. The authors use a hierarchical clustering framework to show that their similarity measure can result in a better clustering when applied to trajectory datasets such as Austrian Sign Language (ASL) [34] when artificial noise is added to the sequences. They recommend to set the threshold ϵ to the quarter of standard deviation of the sequence. A summary of dynamic programming based alignment algorithms is presented in Table 2.1.

Other works have enhanced the edit distance algorithms by adding more computational efficiency. Fast Time Series Evaluation (FSTE) [38] provides a faster way of computing the edit distance based algorithms which are dependent on a threshold such as LCSS and EDR. It partitions the space into grid cells based on the threshold and assigns the sample points of each sequence to their appropriate cell. The matching is then performed in the intersection of the cells associated with each pair. The authors also propose another algorithm, Swale, by assigning a reward for match and a penalty for gap which extends EDR constant gap penalty.

Another type of similarity measures rely on some pattern or property in the time-series. In [39] authors propose to threshold the time-series and use the threshold crossing regions to assess the similarity of sequences. Essentially, they only consider the intervals that contain the sample points with value higher than a pre-defined threshold and compute the distance between the time intervals. This approach shows promising results for certain applications such as environmental air pollution or gene expression data. Chen et al propose a more general work in the same line in [19] by introducing Spatial Assembling Distance (SpADe). SpADe searches for similar pattern in contrasting time-series by scaling and shifting in time and amplitude. Then SpADe measures the similarity only on those similar patterns. The major problem with SpADe is that it requires many parameters such as pattern length, time shift, scale shift etc. to be manually tuned or given. The results also show a performance in par with DTW.

A comprehensive review and experimentation on many of the mentioned similarity measures is available in [21]. The authors empirically show that there exist no absolute winner algorithm and DTW can still be considered as a competitive baseline for similarity measure design. We also consider DTW as our baseline algorithm in this work and show that our methods are able to outperform DTW especially in case of noisy data and even when DTW is coupled with noise removal pre-processing.

Chapter 3

A Motivation: Classification of non-causal EEG signals

3.1 Introduction

The motivation for our work and endeavour to invent an alignment method resilient to noise and non-causality comes from a very interesting research related to cognitive and perceptive sciences. The problem was to discover and identify the underlying behaviour of human brain in response to exposure to human faces. It was known for long in the literature of cognitive science that around 170 ms after the stimuli onset (seeing a face) the perception of faces starts. This result were backed by ERP analysis done in a manual fashion by experts. The recorded brain scans (fMRI or EEG) were averaged and compared to control cases to recover the salient differences that might be associated with face perception. Attempts were made to include statistical analysis to show the significance of these results mathematically [40]. However, the significance of those results were minimal in eyes of researchers in machine learning and statistics community. Therefore, we started applying powerful statistical and signal processing methods on EEG recordings based on a very carefully designed experiment and were able to establish the significance of the marker that happens at approximately 170 ms past the onset (known as N170) and point to other possible markers and regions in the brain that might be of interest for further research. In the following we present those results and the conclusion points to the necessity of a noise and non-causality resilient similarity measure.

EEG signals are very challenging to analyze due to the noisy nature of sampling, cross-talk among different channels and, most importantly, ”artifacts” of routine brain activities such as blinking or breathing. Independent Component Analysis [41], coupled with other dimensionality reduction methods such as the PCA, is often used as a tool to recover the important underlying signals while removing such artifacts. Nevertheless, the unsupervised reduction methods are typically insufficient for identification of features needed for accurate and robust classification of EEG signals as the large variance of input signals does not always warrant its classification significance.

In contrast to unsupervised data-driven feature extraction, common EEG signal analysis approaches focus on fixed dictionaries of wavelet [42], short term Fourier transform [43], or other well-established non-stationary signal representations. However, variability in temporal occurrence of important EEG events, typically exhibited across different subjects and trials, makes the use of such features challenging, requiring temporal alignment. Moreover, fixed dictionary representations, not adapted to data, can be dense, affecting their robustness and making them less attractive for classification settings.

In the Brain-Computer Interface (BCI) literature Common Spatial Patterns (CSP) have shown very promising results [44] as the data-driven means for characterizing temporal patterns that can be used for signal classification. CSP is a linear transformation that maps the original signal into a space where the data shows a high class relative variance. However, CSP's applicability is typically restricted to sparse EEG signals, with well defined temporal boundaries, making such patterns less appropriate for settings of dense signals, such as those in our dataset.

In this work we focus on identification of sparse, data-driven spatio-temporal EEG dictionaries that directly reflect our classification objective. Our goal is not only to identify those patterns as the means of efficient and robust classification but to also assert their relationship with established EEG perceptual signatures such as P100, N170, or P250.

3.2 Predictive models

Our goal is to design robust and accurate predictors of the stimulus classes that can account for high levels of noise/artifacts in the input signal as well as inter-subject variability. We also seek to identify a small subset of features of the EEG signal, the

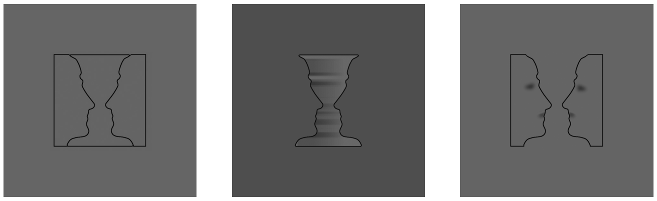


Figure 3.1: Face/vase illustrations

so-called predictive signatures, that contribute to these predictions.

To achieve this goal we focus on a recently proposed family of sparse logistic regression models. Sparse logistic regression models are a class of probabilistic parametric classifiers whose goal is to model the predictive process while minimizing the number of parameters used in this prediction. In particular, consider the binary response variable $y \in \{0, 1\}$, the predictor (feature) vector $\mathbf{X} \in \mathbb{R}^p$. For instance, in the EEG setting y_i can be the class of response (face/vase) while the $k-th$ feature X_k could be the measured EEG signal in channel c at time t. Furthermore, consider the set of N training points $\mathcal{D} = \{(y_i, \boldsymbol{X}_i)\}_{i=1}^N$. The logistic regression model represents the class-conditional probabilities through a sigmoidal function of the linear predictors,

$$\log \frac{Pr(y_i = 1|\boldsymbol{X}, \beta_0, \boldsymbol{\beta}))}{Pr(y_i = 0|\boldsymbol{X}, \beta_0, \boldsymbol{\beta}))} = \beta_0 + \boldsymbol{X}^T \boldsymbol{\beta}. \tag{3.1}$$

The classifier uses the Bayes rule, $f(\mathbf{X}|\beta_0, \boldsymbol{\beta}) = \arg\max_y \; Pr(y|\boldsymbol{X}, \beta_0, \boldsymbol{\beta})$. To find the optimal classifier one can consider a number of objectives, such as the square loss [45] or logistic loss [46], such that loss of prediction on this set of points $\mathcal{D}$ is minimized, while keeping the cardinality k of the utilized features small, $k << p$. For instance, for logistic loss we seek to

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{N} \sum_{i=1}^{N} \log P(y_i|\boldsymbol{X}_i, \beta_0, \boldsymbol{\beta}) \tag{3.2}$$

$$\text{s.t.} \quad card(\boldsymbol{\beta}) \leq k, \tag{3.3}$$

where $card(\cdot)$ denotes the cardinality or the number of non-zero entries in vector $\boldsymbol{\beta}$ (equivalently, the L_0 norm of $\boldsymbol{\beta}$).

Unfortunately, this task is, in general, computationally intractable. One typically instead focuses on a tractable relaxation known as the L_1 or lasso regression [45] (also known as sparse coding), reflected in the Lagrangian:

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{N} \sum_{i=1}^{N} \log P(y_i | \boldsymbol{X}_i, \beta_0, \boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_1 \tag{3.4}$$

where $||\boldsymbol{\beta}||_1$ is the L_1 norm $\sum_{l=1}^{p} |\beta_l|$. Some more recent work has quantified conditions under which the optimization in (3.4) is guaranteed to lead to the same solutions as the original objective (3.3). Nevertheless, in practice the lasso objective leads to models with few non-zero coefficients that focus on those aspects of the feature vector most responsible for distinction between 0/1 classes of inputs. The objective in (3.4) is convex in $\boldsymbol{\beta}$, leading to several efficient gradient based algorithms, see c.f., [46].

Lasso regression focuses on identification of individual features. In many tasks, such as the EEG signal analysis, individual features (values of voltage measured by electrode e at time t) may be insufficiently strong or too variable to lead to robust predictions. Instead, groups of spatio-temporally proximal features may serve as better predictors (e.g, short spatio-temporal signal forms). A typical approach taken in the community is to design such features using fixed dictionaries, such as the Fourier, wavelet, or other bases. In contrast, we seek to find those features directly from data, i.e., identify compact data-dependent spatio-temporal dictionaries of EEG signals. We use Group lasso [47], an extension of lasso, to accomplish this task.

Formally, consider the partitioning of the weight vector $\boldsymbol{\beta}$ into a set of groups $\boldsymbol{\beta}_j, j = 1, \ldots, J$, where each group contains a subset of coefficients $\boldsymbol{\beta}$ (or, equivalently features $\boldsymbol{X}$). Let $||\eta||_K$ be the K norm of vector $\eta \in \mathbb{R}^d, d \geq 1$, $||\eta||_K = (\eta^T K \eta)^{\frac{1}{2}}$ where K is a symmetric positive-definite matrix. The group lasso estimate is defined as the minimizer of

$$\frac{1}{N} \sum_{i=1}^{N} \log P(y_i | \boldsymbol{X}_i, \beta_0, \boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_J) + \lambda \sum_{j=1}^{j=J} ||\boldsymbol{\beta}_j||_{K_j}, \tag{3.5}$$

where

$$\log \frac{Pr(y_i = 1 | \boldsymbol{X})}{Pr(y_i = 0 | \boldsymbol{X})} = \beta_0 + \sum_{j=1}^{j=J} \boldsymbol{X}_{i,j}^T \boldsymbol{\beta}_j. \tag{3.6}$$

In an extension to the lasso regression, Group lasso leads to predictors with few non-zero or *active groups of coefficients*. Here we have considered K to be diagonal.

3.3 EEG Data Collection

Five adults between the ages of 20 and 30 years participated in this experiment. All adults had normal or corrected-to-normal vision, and none had a history of neurological abnormalities.

Our stimulus set consisted of 60 randomly ordered presentations of the face and non-face images shown in figure 3.2. Each trial consisted of stimulus presentation (300ms) and a post-stimulus recording period (1000ms). The inter-trial interval, during which a black fixation cross was presented on a gray background, varied randomly between 1500-2000ms. Participants were required to determine whether each stimulus was a face or a non-face, and responded using two buttons on a button box. Participants were instructed to make their responses as quickly and accurately as possible, and they had 1500ms from stimulus onset to do so.

Electrophysiological Recording and Processing. While participants were performing the above task, continuous EEG was recorded using a 128-channel Geodesic Sensor Net (Electrical Geodesics, Inc.), referenced online to vertex (Cz). The electrical signal was amplified with 0.1 to 100Hz band-pass filtering, digitized at a 250 Hz sampling rate. Data were preprocessed offline using NetStation 4.2 analysis software (Electrical Geodesics, Inc.). The continuous EEG signal was segmented into 900ms epochs, starting 100ms prior to stimulus onset. Data were filtered with a 30Hz low-pass elliptical filter and baseline-corrected to the mean of the 100ms period before stimulus onset. NetStation's automated artifact detection tools combed the data for eye blinks, eye movements, and bad channels. Segments were excluded from further analysis if they contained an eye blink (threshold $\pm 70\mu V$) or eye movement (threshold $\pm 50\mu V$). In the remaining segments, individual channels were marked bad if the difference between the maximum and minimum amplitudes across the entire segment exceeded $80\mu V$. If more than 10% of the 128 channels were marked bad in a segment, the whole segment was excluded from further analysis. If fewer than 10% of the channels were marked bad in a segment, they were replaced using spherical spline interpolation. For the subsequent data analysis we retained signal ranges from 0ms (stimulus onset) up to 700ms.

Test cases	L1 Logistic Regression					Group Lasso				
	full	50-100	100-200	200-300	$\geq$ 400	full	50-100	100-200	200-300	$\geq$ 400
subject 1	82.40	70.04	80.14	78.48	57.91	**88.11**	66.45	79.43	68.37	55.77
subject 2	**98.26**	69.10	94.10	86.11	50.00	92.53	73.26	93.92	77.60	50.00
subject 3	**86.88**	77.16	86.11	82.72	50.00	73.46	54.78	85.49	70.06	50.00
subject 4	84.78	74.90	82.02	80.83	67.00	**87.94**	66.01	82.81	79.45	64.23
subject 5	80.74	64.64	**91.79**	59.64	58.93	**91.43**	60.71	87.89	59.29	63.93
Average	**86.61**	71.17	**86.83**	77.56	56.77	**86.69**	64.24	85.91	70.95	56.77

Table 3.1: Across subject AUCs for L1 logistic regression and group lasso, in percentage points, for different time ranges.

3.4 EEG Data Analysis

In our experiments we used L1-norm logistic regression and group lasso described in section 3.2, to design classifiers and spatio-temporal dictionaries for the purpose of predicting the class of visual stimulus to which our subjects were exposed. We used the methods of [46] and [48] to train the two classifiers.

In the data pre-processing stage we filtered the EEG signals by an FIR band-pass filter 8-14 Hz to retain a portion of the alpha range of brain signals. We then normalize each channel independently and windsorize 5% of the peak. Each processed signal forms a $129 \times 700 = 90,300$ dimensional feature vector $\mathbf{X} = [X_{i,t}]$, $i = 1, \ldots, 129$, $t = 1, \ldots, 700$. For the group lasso setting we formed spatial groups of size five for each channel and its four nearest spatial neighbors. Temporal groups were set to fixed size of 10 samples, set at every 5th sample (i.e., 1-10, 5-15, etc.)

As a baseline classifier, we also use a k-nearest neighbor (kNN) predictor which transfers labels from the training instances using majority voting to the query instance. For kNN we used the following heuristic to remove the outliers. On training data we computed the median of each class as its corresponding center. We classified the training points using those centers with k equal to 90% of the cluster size. We then removed those samples which were not classified correctly and found the centers again. This procedure was repeated several times. The resulting centers were able to classify more than 80% of the training data correctly, which we deemed sufficient to prevent overfitting. We used L1 norm as the measure of distance between points.

To assess the ability of classifiers to identify established signal patterns typically used in visual perception, such as the P100, N170, P250, we contrast the three models constructed over the full range of temporal data (0-700ms) with those constructed with data limited to specific temporal ranges. We chose the ranges of [50ms-100ms], [100ms-200ms], [200ms-300ms], and over 400ms.

We focused on the typically more challenging across-subject classification task where we test the ability of classifiers to generalize across different subjects. To accomplish this we train our classifiers on all but one subject and test the model's performance on the EEG sequences of the remaining subject. We report the AUC (area under ROC curve) scores for all classifiers as a measure of their classification effectiveness.

3.5 Experimental Results

Table 3.1 summarizes predictive performance of the two classifiers across the subjects, as well as the average performance. Results for the baseline kNN approach were substantially worse across all temporal ranges. Average AUC scores were 51.63%, 54.79%, 56.16%, 63.51%, 58.65%, starting with the full all the way to ≥ 400 range. Without the aforementioned outlier removal technique in Section 3.4 the AUC scores downgrade to 48.29%, 47.26%, 51.46%, 54.8%, 52.70%, respectively.

The compound AUC measure suggests that both L1 and the group lasso can give effective prediction for across-subject predictions but varies significantly across subjects. Responses of some subjects, such as 2 & 3, can be effectively predicted using isolated spatio-temporal features of the L1 regression. On the other hand, responses of subjects 1, 4, and 5 are substantially better predicted using the group model, which signifies the importance of collective activity in select spatio-temporal regions. On average, simultaneous use of features across the full temporal range of 0-700ms leads to best performance, but this is also observed if the temporal range is restricted to 100ms-200ms with the L1 prediction, closely followed by the same restricted range with group lasso. This is clearly in agreement with the known role of the N170 potential in detection of human faces. No other isolated temporal range has lead to similar level of classification performance.

We next examined the spatio-temporal localization of features selected by our sparse predictors, for the across-subject setting. This is illustrated in Figure 3.2. Results are shown for average models (over the five subjects) and, as an exemplar, for one specific subject (Subject 1). The two left most columns represent spatial locations and (relative) intensities of non-zero features, accumulated over the full temporal range. Namely, if $\beta_{i,t}$ is the weight of the (processed/filtered) response $X_{i,t}$ of channel i at time t, the weights displayed correspond to $\beta_i = \sum_t \beta_{i,t}$. The first of those columns indicates spatial locations of features present when subjects are exposed to 'face' stimuli, while the next column shows the same for 'vase' stimuli. The spatial locations show consistency both across subjects and across the two models (L1 and group lasso). Group models induce selection of larger spatial groups of electrodes, based on our definition of spatial neighborhoods. In general and as expected, L1 models reduce this spread. It is interesting that the sensors selected by our classifiers mostly lay in the occipital and temporal areas of the brain, the visual regions that have been implicated in object recognition.

The right-most column in Figure 3.2 displays the temporal distribution of weights, accumulated over all electrodes: the cumulative weight is $\beta_t = \sum_i \beta_{i,t}$, computed separately for all positive (face/blue) weights and negative (vase/red) weights. Again, we observe significant amount of consistency across five subjects. Moreover, the temporal placement of the selected weights shows interesting correlation with known indicators such as the N170 and P250, with the weight around N170 showing very strong peaks. We also observe a considerable concentration of 'face' weights around 400ms, which can be associated with N400. The third concentration around 600ms can be seen as a continuation of the N400. The temporal position of the 'vase' features exhibits very strong peaks slightly before the 'face' signals, around 120-130ms.

In the case of the L1 model, which enforces no temporal smoothness (grouping) of selected features, we also observe strong peaks in some subjects in the range of 50ms-100ms. Not unexpectedly, the temporal signatures show more variability than the corresponding temporal signatures of group lasso model. It should finally be noted that in the case of L1 model and subject 5 we observe temporal signatures that deviate from those in all other cases. This, together with the relatively low AUC scores for L1 on subject 5 and the high scores for the same subject with restricted temporal

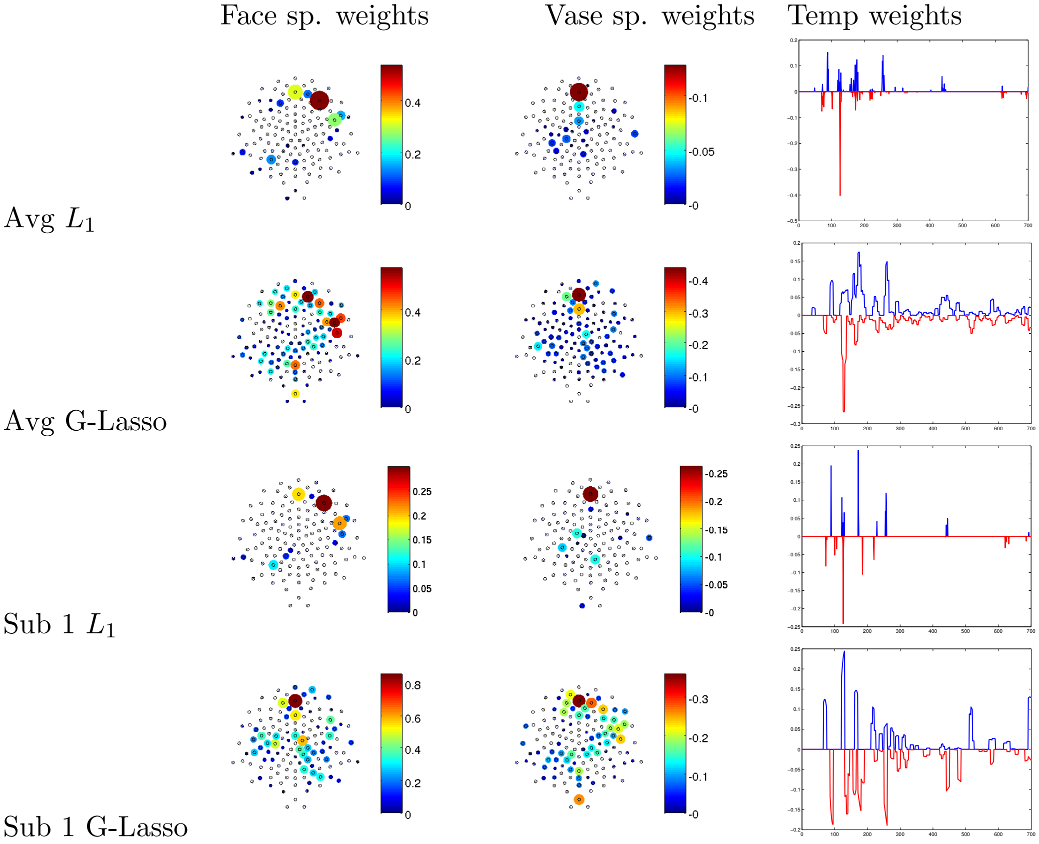


Figure 3.2: Average and Subject 1 spatial and temporal distribution of positive (face/blue) and negative (vase/red) weights estimated by L1 logistic regression and group lasso models. Top of each sensor map corresponds to the occipital brain region.

range (100-200), suggests the existence of spurious (outlier) features at fine temporal scales. However, group lasso successfully eliminated these outliers by requiring collective excitation over larger temporal windows.

It is important to stress that under either of the two modeling approaches our data-driven models recovered specific stimulus signatures that are very closely associated with, but not identical to, established indicators. This is significant from two perspectives. First, it validates the analysis methodology presented here by showing that one can pick out accepted spatio-temporal correlates in a completely agnostic fashion. This is thus a potentially general methodology for use in identifying EEG markers for tasks that hitherto have not been associated with any distinct ERP components. Second, and more specifically, to the extent our results differ from the discrete ERP face correlates, they suggest that additional temporal epochs, besides just the 170 or 250 ms points, might carry information regarding face perception. Future investigations can probe whether this additional information is associated with aspects of face perception beyond just labeling a pattern as a face.

3.6 Conclusion

In this work we showed that the brain signals recorded from subject when exposed to face stimuli is statistically significantly different from the control signal where the subjects were exposed to vase stimuli. We were also able to recover the salient time markers and sensors contributing to the distinction of the two classes. Not only we were able to assert the traditional markers such as N170 but we also pointed possibly new time markers which encourages further research in this area.

All great points mentioned above are true but despite our numerous attempts we did not have much success in improving the recognition results any further. The reason for that, relies on the fact the signals are not aligned. Even the scans from the same subjects may not be aligned. This cause the distribution of futures to be noisy and thus the regressor will fail to capture and recover the appropriate coefficient. The solution is to align all sequences. The problem however, is that the traditional alignment algorithms such as Dynamic Time Warping c.f. [23] are extremely susceptible to noise while EEG signals carry a significant amount of noise. Moreover, EEG signals tend to be non-causal [9]. This means that the order of time points might differ from one signal to another at least locally. However, the time monotonicity is a key assumption in all alignment algorithms.

These properties of EEG signals motivated us to design an algorithm for alignment that is noise-resilient and insensitive to local reordering of time-points (local non-causality). We later show that the methods built upon this idea is applicable to general time-series alignment and significantly outperforms all alignment algorithms when the signal is severely noisy and/or non-causal.

Chapter 4
Isotonic Canonical Correlation Analysis

4.1 Introduction

As we discussed in Chapter 2, the problem of sequence alignment arises in many fields of science as a consequence of dealing with data that does not live in fixed dimensional Euclidean spaces. In this Chapter we focus on computer vision problems. In computer vision, sequence alignment is an important first step used to solve problems such as the human activity analysis and recognition, c.f., [4]. The alignment can be used to establish a measure of similarity between two sequences of video frames or motion capture data, which can be subsequently employed for sequence classification or clustering [49].

As we discussed before, a traditional way to address the alignment problem between two sequences relies on Dynamic Time Warping (DTW). DTW is typically solved using a combinatorial dynamic programming algorithm that searches for a globally optimal warping path, mapping the domain of one sequence onto the other. DTW and its derivatives have shown great success in many practical alignment applications [21]. In practice the unconstrained warping of a generic DTW often fails to yield reasonable and robust results. Imposing constraints on the feasible warping paths has empirically shown to improve the classification performance [50, 24, 51]. For instance, the Sakoe-Chiba band constraint [24], which restricts the maximum deviation of matching slices from the diagonal by $p\%$ of the sequence length, can result in substantially improved alignments depending on the choice of p. Recently, [51] proposed an adaptive band approach that estimates function spaces of time warping paths, removing the need for a fixed p. In this setting motion class-specific warping-path constraints are learned for each class that reflect the warping variations of samples within it. Nevertheless, imposing a proper set of constraints on DTW is still a challenging problem.

Segmental Alginment: One additional property of DTW-based alignment is that it assumes pairing of individual data points: a sample at time t_x in sequence x is typically aligned with only one other sample at time t_y in sequence y. In many practical applications it may be more desirable to establish pairing between groups of points: associating a temporal segment $\mathbf{t}_x = [t_{x,1}, \ldots, t_{x,2}]$ to another segment of the contrasting sequence, $\mathbf{t}_y = [t_{y,1}, \ldots, t_{y,2}]$. The segment pairing can reflect alignment of e.g., segment sufficient statistics instead of the raw sample values, making the comparison more robust to individual sample differences. For example, in mocap data segment-level alignment can be less susceptible to individual subject differences while performing a certain motion or activity. Alignment on the segment level can also be justified as a way of comparing the local segment densities, in view of the Hilbert space embeddings of probability distributions [52]. Additionally, point-to-point pairing deems DTW to be very sensitive to noise. While pairing convex combinations of the segments of the two contrasting sequences can impose additional filtering on data leading to a more robust alignment.

A natural way of solving the alignment problem is to find the aligned subspace (manifold in general) where the correlation of two sequences is maximized. Canonical correlation analysis (CCA) provides the necessary means for finding such a subspace between a pair of random variables. However, CCA in its original formulation does not respect the critical *monotonicity* property of any temporal alignment, which prevents arbitrary permutations of time indexes (e.g., self intersection of time, mapping $t_{x,1} \rightarrow t_{y,2}$ and $t_{x,2} \rightarrow t_{y,1}$ when $t_{x,1} < t_{x,2}$, $t_{y,1} < t_{y,2}$). To address this problem [31] proposed to formulate CCA of two vectors as a regression problem with a proper hyperbolic basis for the coefficient vector, guaranteeing monotonicity and the isotonic character of this solution. Their approach, however, does not immediately extend to multivariate time series and also does not explicitly model the segment alignment. A CCA-based formulation was also used in [4] but the alignment was accomplished using traditional DTW.

In this Chapter we present a novel approach based on an isotonic extension to CCA to tackle the problem of sequence alignment. Unlike the traditional CCA, we introduce alternative constraints that guarantee monotonicity in the projected alignment space. We show that these constraints are of quadratic nature, similar to the traditional CCA normalization constraints. This set of constraints is supplemented with non-negativity and norm-1 normalization constraints, which together allow alignment of segments and the Hilbert density embedding interpretation. We then present an efficient solution to the isotonic CCA optimization based on iterative coordinate descent and non-negative least squares. The performance of the isotonic CCA alignment is evaluated in synthetic experiments and on a MoCap-based activity recognition task, where it is contrasted to traditional alignment approaches. The results demonstrate that the new method can improve recognition accuracy and exhibit resilience to noise in cases where traditional sample-based DTW alignments are prone to failure.

4.2 Isotonic CCA Model

In this section we formulate the problem of aligning two sequences X and Y as a general Isotonic CCA task. We first introduce the notation and the unconstrained CCA objective and follow by developing the necessary set of constraints that guarantee monotonicity and allow segment-based alignment.

Let the two multivariate sequences X and Y be represented as matrices of size $N \times T_x$ and Y is a $N \times T_y$, respectively. The unconstrained CCA alignment objective is:

$$(W_x, W_y) = \arg \min_{W_x, W_y} \frac{1}{NT} \|XW_x - YW_y\|_F^2, \tag{4.1}$$

where W_x and W_y are the linear warping matrices of size $T_x \times T$ and $T_y \times T$, respectively. The role of these warping matrices is to map the two sequence time-scales into a common space (of dimension T here) where the sequences become most similar (in the L2 sense).

Traditional CCA, c.f. [30], imposes orthonormality quadratic constraints $W_x^T cov[X] W_x = I$, $W_y^T cov[Y] W_y = I$, $W_x^T cov[X, Y] W_y = 0$ to find the set of projections W_x and W_y. However, such constraints do not guaranty monotonicity in the projected space, namely that

$$t_1 \leq t_2 \Rightarrow w_x(t_1) \leq w_x(t_2). \tag{4.2}$$

Here $w_x(t)$ is the continuous warping function and W_x is the sampled version of $w_x(t)$ where $W_x(i, j) = 1 \Leftrightarrow w_x(j) = i$ indicates that i-th sample of sequence X is mapped onto the j-th sample of the canonical time coordinate. In the next section we propose an alternative constraint set of monotonic constraints to replace the traditional CCA orthonormality constraints.

4.2.1 Monotonicity Constraints

To introduce the new monotonicity constraints we next consider the case when W_y is fixed. We denote by $Z = YW_y$. The same considerations are valid when both W_x and W_y, although the above assumption simplifies the exposition.

Consider the problem of monotonically aligning an $N \times T_x$ matrix X to a $N \times T$ matrix Z. For this alignment to respect monotonicity constraints we define the optimization as follows:

$$\arg\min_{W} \frac{1}{NT} \|XW - Z\|_F^2 \tag{4.3}$$

subject to

- Monotonicity

$$w_k^T(\mathbf{1} - Cw_l) = 0, \quad k > l \tag{4.4}$$

$$w_k^T(\mathbf{1} - C^T w_l) = 0, \quad l > k \tag{4.5}$$

- Simplex (normalization and non-negativity)

$$W^T\mathbf{1} \;=\; 1 \tag{4.6}$$

$$W \;\geq\; 0 \tag{4.7}$$

Here w_i denotes the ith column of W. $\mathbf{1}$ is a $T \times 1$ vector of all 1's. Constraint (4.7) is non-negativity on per-element basis. C is a $T \times T$ *cumulative* operator defined as

$$C = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \tag{4.8}$$

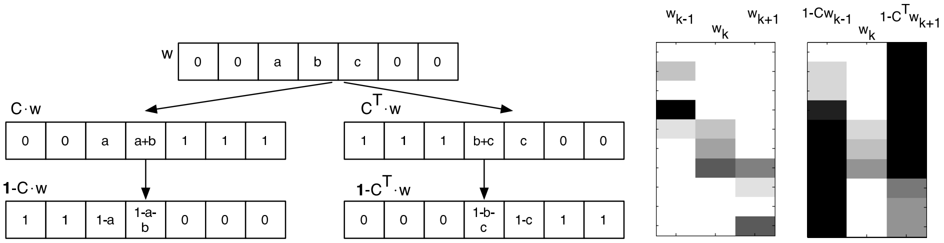


Figure 4.1: Effects of operators $w_k^T(\mathbf{1} - Cw_{k-1})$ and $w_k^T(\mathbf{1} - C^T w_{k+1})$ (row vector format). The first operator "forces" the leading elements of w_k to be zero, up before the last nonzero element of w_{k-1}. The second operator constrains the trailing elements of vector w_k to zero, past the first element of w_{k+1}. The center panel shows an example of a non-negative matrix W that satisfies those constraints. The rightmost panel shows results of applying the constraint operators discussed above on the first and the last column of W. w_k can be non-zero wherever both Cw_{k-1} and Cw_{k+1} are identically equal to 1. White = 0, black = 1.

The idea behind the definition of monotonicity constraints (4.4,4.5) is as follows. Consider the k-th column of matrix W, w_k. Constrains (4.6,4.7) ensure that each column vector is non-negative and sums to one (i.e., a stochastic vector). Now consider the effect of operator $\mathbf{1} - Cw_{k-1}$, premultiplying the previous row of W by C and subtracting this result from 1. Premultiplication by C constructs the CDF (cumulative score) of w_{k-1}. Subtraction from 1 constructs the "complement" of this CDF. This is illustrated in Figure 4.1.

The outcome of applying these operators is that the any pair of columns of W has to be "orthogonal", in the sense illustrated in Figure 4.1. The constrains (4.4) requires all elements of column w_k to be zero for all indexes before the index of the last non-zero element in the previous columns. Conversely, (4.4) states that any element of w_k has to be zero past the first non-zero element of the subsequent columns. Note that the non-negativity of W, together with normalization, is crucial for these interpretations to hold. These constrains will result in a generalization of traditional DTW constraints.

We also note that (4.6) implies that (4.4) and (4.5) can simplified to

$$w_k^T C w_{k-1} = 1, \; w_k^T C^T w_{k+1} = 1, \; \forall k = 2, \ldots, T-1. \tag{4.9}$$

Notice that all pairwise constraints have been reduced to the smaller set of $T-2$ constraints. This is because "orthogonality" of w_k and w_{k-1} automatically implies "orthogonality" of w_k and w_l, $\forall l < k$. Similar argument holds for the other constraint.

4.2.2 Isotonic CCA Objective

The isotonic CCA objective, which satisfies monotonicity constraints, can now be written as:

$$(W_x^*, W_y^*) = \arg\min_{W_x, W_y} \frac{1}{NT} \|XW_x - YW_y\|_F^2 + \tag{4.10}$$
$$\lambda_1 \|W_x\|_F + \lambda_2 \|W_y\|_F$$

s.t.

$$w_{X,k}^T C w_{X,k-1} = 1,\ w_{X,k}^T C^T w_{X,k+1} = 1, \quad 1 < k < T$$
$$w_{Y,k}^T C w_{Y,k-1} = 1,\ w_{Y,k}^T C^T w_{Y,k+1} = 1, \quad 1 < k < T$$
$$W_X^T \mathbf{1} = \mathbf{1}, W_X \geq 0 \tag{4.11}$$
$$W_Y^T \mathbf{1} = \mathbf{1}, W_Y \geq 0 \tag{4.12}$$

Unlike traditional CCA, we do not require the cross-term constrains ("orthogonality" of W_X and W_Y). Regularizing the objective with L2-norm of the parameters, given Eq. 4.11 and Eq. 4.12, increases the number of non-zero coefficients within time segments. The latter will lead to higher resilience to noise by "spreading" the weights over segments instead of focusing on individual samples.

4.3 Optimization of Isotonic CCA Objective

Traditional CCA can be solved in closed form using a generalized eigenvalue-eigenvector analysis, see e.g., [30]. The isotonic CCA, on the other hand, contains non-negativity constraints in addition to a (smaller number) of quadratic and linear (normalization) constraints. We thus follow an alternative approach of *alternating least squares* to solve the optimization at hand. In this formulation CCA objective can be solved by holding one projection matrix fixed while optimizing the other using traditional least squares algorithms, as indicated by the form of (4.3). This procedure is repeated by alternating the roles of the two warping matrices, until convergence.

Each individual least-squares problem of form (4.3) contains quadratic constraints outlined in the previous section. The resulting quadratically constrained quadratic program (QCQP) also contains the non-negativity constraints, leading to a non-negative

QCQP. However, if all w_k but one are assumed fixed, the problem turns into a nonnegative least squares problem with linear constraints.

Let W_{-k} denote the W matrix in (4.3) with the k-th column removed. The optimization problem for the k-th column of W becomes

$$\arg\min_{w_k} \frac{1}{N} \|Xw_k - z_k\|^2 \tag{4.13}$$

subject to

$$A(W_{-k})w_k = \mathbf{1}$$
$$w_k \geq \mathbf{0}$$

where

$$A(W_{-k}) = \begin{bmatrix} w_{k-1}^T C^T \\ w_{k+1}^T C \\ \mathbf{1^T} \end{bmatrix}. \tag{4.14}$$

The first two constraints of $A(W_{-k})$ are satisfied when the parts of w_k corresponding to 1s in $w_{k-1}^T C^T$ and $w_{k+1}^T C$ are zero:

$$w_k = \begin{bmatrix} \mathbf{0} \\ \tilde{w_k} \\ \mathbf{0} \end{bmatrix} \tag{4.15}$$

Hence, the problem can be now be formulated in terms of the remaining nonnegative portion $\tilde{w_k}$ as

$$\arg\min_{\tilde{w_k}} \frac{1}{N} \|\tilde{X}\tilde{w_k} - z_k\|^2 \tag{4.16}$$

subject to

$$\begin{aligned} \mathbf{1}^T \tilde{w_k} &= 1 \\ \tilde{w_k} &\geq 0 \end{aligned} \tag{4.17}$$

where $\tilde{X}$ is the submatrix (columns) of X corresponding to $\tilde{w_k}$.

This new problem is a case of nonnegative least squares with equality constraints. In particular, following [53] we define

$$\mathbf{1^T} = \mathbf{USV^T} \tag{4.18}$$

Here

$$U = 1$$

$$S = \left[\begin{array}{cccc} \sqrt{n} & 0 & \dots & 0 \end{array} \right]$$

$$V = \left[\begin{array}{cccc} 1/\sqrt{n} & 1/\sqrt{n} & \cdots & 1/\sqrt{n} \\ 1/\sqrt{n} & a & \cdots & -b \\ \vdots & -b & \ddots & -b \\ 1/\sqrt{n} & -b & \cdots & a \end{array} \right] = \left[\begin{array}{cc} v_1 & V_{-1} \end{array} \right] \tag{4.19}$$

Let

$$\tilde{w_k} = K \left[\begin{array}{c} y_1 \\ y_2 \end{array} \right] = V/S_1 \left[\begin{array}{c} y_1 \\ y_2 \end{array} \right] \tag{4.20}$$

Then

$$1^T K = \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \end{array} \right] \tag{4.21}$$

$$\tilde{X} K = \left[\begin{array}{cc} \tilde{X}1/n & \tilde{X}V_{-1}/\sqrt{n} \end{array} \right] \tag{4.22}$$

and the optimal solutions are

$$y_1^* = 1 \tag{4.23}$$

$$y_2^* = \arg\min_{y_2} \|\tilde{X}V_{-1}/\sqrt{n}y_2 - (z_k - \tilde{X}1/n)\|^2 \tag{4.24}$$

subject to

$$V_{-1}/\sqrt{n}y_2 \geq -1/n \tag{4.25}$$

To convert the above LSI problem to non-negative least squares, we have to first convert it into a least distance problem (LDP) and the solve the LDP using non-negative least squares. Now we can apply the exact same steps in [53] by setting $E = \tilde{X}V_{-1}/\sqrt{n}$, $f = (z_k - \tilde{X}1/n)$, $G = V_{-1}/\sqrt{n}$ and $h = -1/n$.

4.3.1 Initialization and complexity analysis

Because the program in Eq. 4.10 is non-convex due to quadratic equality constraints, initialization can be critical. A reasonable initialization, suggested by methods such as [4] can be based on the DTW path. In our case standard DTW initialization needs to be modified to account for the segmental alignment imposed by our objective. This can, for instance, be accomplished by averaging successive columns of the corresponding DTW matrices W_{DTW}, e.g., $w_i = \frac{1}{K}\sum_{k=0}^{K-1} w_{DTW,K\cdot i+k}$. The time complexity of the overall isotonic CCA optimization depends the embedding dimension, T, and the average time segment length K, as $O(TK^3)$. The values of K and T are usually dependent on the time length of the sequences, T_x and T_y. That is, $K \propto \rfloor \frac{\max(T_x,T_y)}{T}$ and for instance in our experiments $T = \lfloor \frac{\min(T_x,T_y)}{2} \rfloor$. Therefore in this case, the time complexity will be dependent only on T_x and T_y, i.e. $O(\frac{\max(T_x,T_y)^3}{\min(T_x,T_y)^2})$.

4.4 Experimental Results

In this section we demonstrate the utility of IsoCCA through experiments conducted on synthetic data as well as 3D human motion capture (MoCap) time series, similar to [4]. MoCap data depicts trajectories of joint angles of human subjects that perform various periodic and aperiodic activities. We focus on classification tasks as the measure of alignment quality in both settings. We assessed the noise robustness of the proposed approach and investigated the effect of parameters and initialization on the algorithm performance. In all experiments we set $T = \lfloor \frac{\min(T_x,T_y)}{2} \rfloor$ and $\lambda_1 = \lambda_2 = 0.5$, unless otherwise noted.

4.4.1 Synthetic Data

To investigate the properties of IsoCCA and contrast it with DTW, we generated a synthetic dataset consisting of 10 examples each from two classes of univariate sequences. The first class is a segment of a sinusoid while the second class is a rectangular signal, both randomly embedded in segments of Gaussian noise. Heterogeneous noise is generated from five Gaussian processes with means $\mu \in [0, 5]$ and variances $\sigma^2 \in [0, 10]$ chosen uniformly at random. Two examples of sequences from the sinusoid class are depicted in Figure 4.2. IsoCCA is initialized from DTW by averaging blocks of DTW projection matrices W_x and W_y, as described in Section 4.3.1.

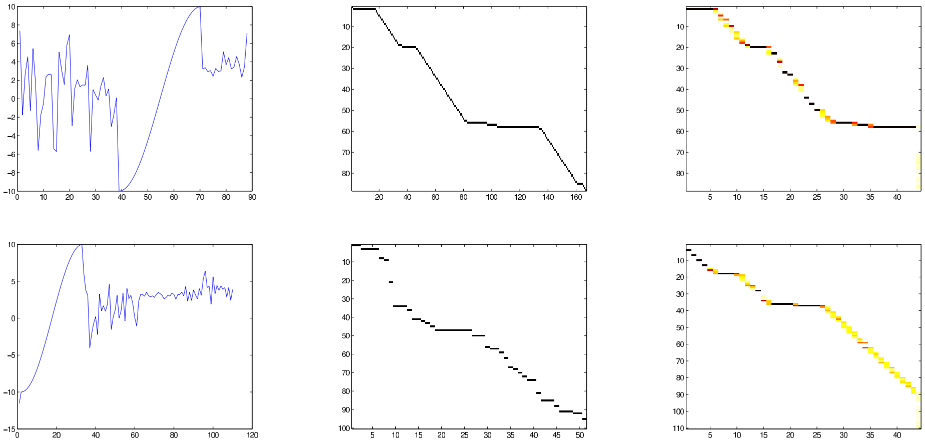


Figure 4.2: From left to right: sample sequences, DTW and IsoCCA warping matrices.

Using a 1-NN classifier and leave-one-out cross-validation query sequences were classified in one of the two classes. This yielded recognition rates of 60.00% (12 out of 20) for DTW vs. 90.00% (18 out of 20) reported by IsoCCA. High levels of noise led to spurious DTW alignments, as illustrated in Figure 4.2. On the other hand, IsoCCA was able to more accurately capture the alignment of the essential signals parts, while discarding the noise, as alluded to by the shown alignments.

4.4.2 Motion Capture data

We selected 62 sequences containing more than 40000 frames of 8 different actions from the CMU MoCap data [54]: walking, runing, boxing, jumping, marching, dancing, sitting down and shaking hands. Each class contains 7, 10, 8, 6, 10, 10, 7 and 4 sequences, respectively. Classes were selected with actions performed by different subject. The dimensionality of data is reduced from 62 to 10 using PCA while keeping 99.8% of the energy. Figure 4.3 shows examples of selected videos and their corresponding warping matrices.

We compared our method to DTW, CTW [4], best unconstrained matching (Hungarian algorithm), and standard CCA. 1-NN is used as the classifier to find the closest sequence to any given query in a leave-one-out setting. In DTW Sakoe-Chiba constraint with $\rho = 15\%$ is imposed to improve its performance in classification. The best

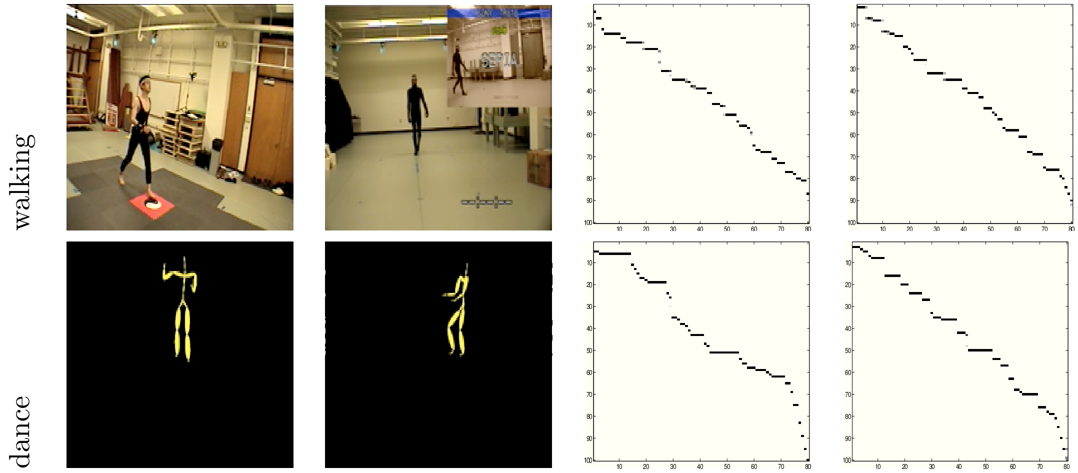


Figure 4.3: Example actions and their corresponding IsoCCA warping matrices.

unconstrained matching imposed no monotonicity constraints, as was the case with the standard CCA. The overall accuracies are shown in Table 4.1.

Method	IsoCCA	DTW	CTW	CCA	Hungarian
Accuracy	**87.10%**	80.65%	54.05%	45.16%	43.55%

Table 4.1: MoCap recognition rates.

Applying CTW to the MoCap recognition problem in the best case yields a recognition rate lower than the baseline DTW. A ostensible benefit of CTW is the pairwise spatial alignment. However, this is not a concern in MoCap data with known correspondences. In our setting CTW produces similar distance scores across ranges of motion, indicating over-fitting. The global PCA used here (same spatial embedding for all elements in the dataset) is less prone to differences in individual data-pairs. To improve performance of CTW one would have to carefully control the spatial projections, perhaps using within-class regularization or some global regularization constraints. However, we were unable to find such a setting in our experiments. Furthermore, we compared our method with parametric warping proposed in [55] for which the recognition rate was 25%. The confusion matrix for IsoCCA is presented in Table 4.2. Throughout the experiments we have used DTW alignment path as the initialization point.

	walk	run	boxing	jump	Marching	salsa dance	sit	shake
walk	100	0	0	0	0	0	0	0
run	0	100	0	0	0	0	0	0
boxing	0	0	100	0	0	0	0	0
jump	0	0	0	100	0	0	0	0
Marching	0	0	0	0	100	0	0	0
salsa dance	0	0	0	0	0	60	40	0
sit	0	0	43	0	0	0	57	0
shake	0	0	0	25	0	0	0	75

Table 4.2: Confusion matrix for IsoCCA(in percentage points)

Noise resilience analysis

To assert robustness to noise we added two types of noise to the clean MoCap data. We have compared DTW and IsoCCA in case of additive Gaussian noise and sparse noise spikes. The noise process in the case of additive Gaussian noise is $N(0, p\sigma_i)$, added to i-th feature with σ_i the standard deviation of the feature and $p \in [0, 1]$. In case of spike noise, we have randomly added values drawn from the normal process, $N(0, p\sigma_i)$, to randomly chosen time points of each feature. The number of noisy time points is not more than 5% of the length of the time-series, spread uniformly over the full time-span. Figure 4.4 depicts DTW and IsoCCA recognition rates in presence of different levels of noise. Figure 4.4 indicates that IsoCCA outperforms DTW a margin that increases as the noise level grows. We will shortly show that the algorithm parameters can change the recognition performance drastically in case of noisy data.

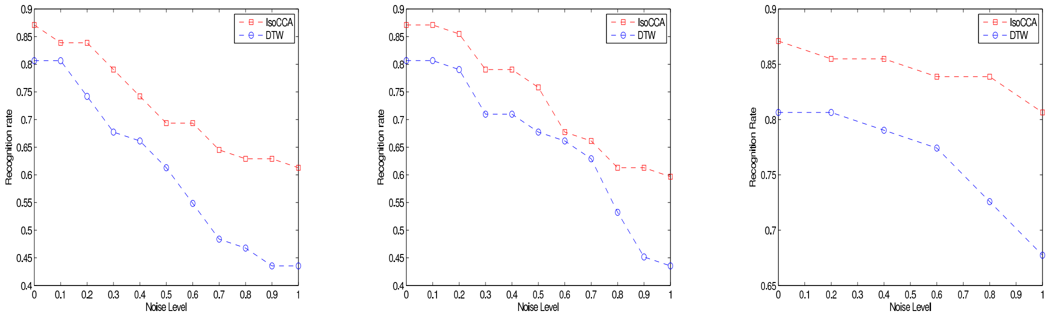


Figure 4.4: From left to right: noisy (additive Gaussian) query and clean training set, query and training are both noisy (additive Gaussian), noisy (sparse spikes) query and clean training set.

4.4.3 Parameter sensitivity

An important aspect of the proposed approach relates to the choice of the common embedding dimension T and the "spreading factors" $\lambda = \lambda_1 = \lambda_2$,the L2-norm regularization coefficients. Intuitively, smaller T would result in alignments of longer time segments. Larger λ, as discussed earlier, increases the number of non-zero elements in warping matrices. Figure 4.5 shows the dependence of the recognition rate as a function of T in the range 1 to $\lfloor \frac{\min(T_x,T_y)}{2} \rfloor$, in the presence of different levels of additive Gaussian noise. Interestingly, with no noise added, changing T does not significantly impact the recognition rate. As the noise strengthens, smaller T results in improved recognition (down to a level), indicating the importance of segmental alignments.

In the experiments we found the algorithm largely insensitive to λ, with the recognition rate slightly improved for higher λ in the presence of noise. Nevertheless, the L2-regularization is critical for yielding a stable objective.

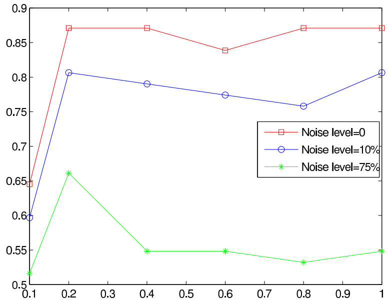


Figure 4.5: Effect of changing common dimension T. Horizontal axis shows proportion of $T_{max} = \lfloor \frac{\min(T_x,T_y)}{2} \rfloor$

4.5 Conclusions

In this Chapter we presented an alignment algorithm based on isotonic CCA which linearly maps two sequences to a common subspace such that the non-decreasing monotonicity in time is preserved. In addition, the alignment approach naturally fosters alignments of sequence segments instead of individual samples. We presented a solution for the isotonic CCA based on non-negative least squares. Our experimental results

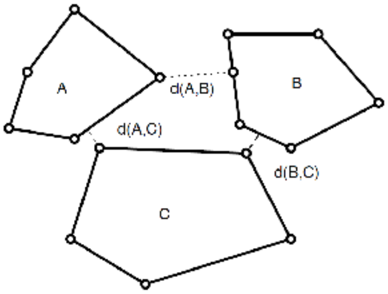


Figure 4.6: IsoCCA works by finding nearest neighbour distance between convex hulls of the segments in contrasting sequences. However, nearest neighbour is not proper metric. That is $d(A,C) + d(B,C) \ngeq d(A,B)$.

show that the segment-based alignment of IsoCCA can be beneficial in cases when high levels of noise can reduce robustness of traditional DTW alignments.

The proposed framework is general and can be extended to simultaneous alignment of multiple sequences, using generalizations of CCA from pairs to sets of datapoints. Despite promising preliminary results, computational complexity of the proposed solution and its dependence on initial conditions may be of concern. Additional consideration is necessary to improve the algorithmic efficiency and scalability. Additionally, the retrieved segments tend to be very short and unrealistic. There are also many jump in the the alignment path.

The main problem with IsoCCA is that the proposed framework does not provide a proper metric between the segments. The reason for that lies in the fact that IsoCCA works by effectively finding the closet points of the convex hulls of the two segments of points. This results in a non-metric because the triangular inequality does not hold (Figure 4.6). Moreover in the case of overlapping convex hulls, their distance is zero even though the size of the common area can be very small resulting in unnecessarily small segments. The IsoCCA objective (4.10) does not guarantee to correspond every point in one sequences with one or more points in the other. In other words, the alignment path produced by IsoCCA might have jumps. All these properties result in having unrealistic segments. IsoCCA works well as a similarity measure but the segmentation can be improved.

Chapter 5

Segmental Pair-HMM

5.1 Introduction

In Chapter 4 we introduced our first segmental alignment algorithm, IsoCCA. Even though IsoCCA is resilient to noise, it is sensitive to the choice of initial point and does not provide a good segmentation. In this Chapter we present a new method that produces much better and coherent segments. We expect that better segmentation must result into a better simialrity measure.

In [5] author proposes a method relevant to the approach that we will be presenting in this Chapter. Ryoo proposes to find the best matching segments of the two sequences based on a probabilistic model. However, the algorithm does not handle gaps/insertions and, hence, does not consider a complete alignment model. Moreover, the author suggests empirically fixing all segment lengths, with the approach lacking clear means to handle data-driven segments. In practice, however, variable and data-adapted segments result in more robust alignments.

In this Chapter we propose a segmental alignment framework based on a probabilistic model and investigate its properties and robustness against noise in the context of sequence classification. The new contributions are:

- We suggest a distance metric based on average pair-wise distances suitable for measuring similarity between two segments , aimed at segmental sequence alignment.

- Based on the proposed distance metric we develop a probabilistic alignment model by extending the pair-HMM formalism.

- We propose a relaxation to the original model and use bounding techniques to reduce the computation time.

Through extensive experiments we show that the proposed method can lead to improved classification results on benchmark sequence classification tasks, classification of non-causal EEG signals, and recognition of activities from human motion data. This proposed approach is particularly resilient to noise where other similar approaches fail.

This Chapter is organized as follows: in Section 5.2 we construct our segmental metric. In Section 5.3 the proposed model is discussed in detail. Section 5.5 introduces the relaxed model for reduced computational time. In Section 5.6 experimental results is presented followed by Section 5.7 that concludes this Chapter with the discussion of our findings and some suggestions for future work.

5.2 Segment Matching Metric

In some applications, as illustrated in Chapter 1, one is interested in matching unordered small segments of points. This naturally leads to matching two unordered sets of points where permutation is not a matter of concern. In addition to insensitivity to permutation, we seek to find a distance metric the suppresses the noise and is efficient to compute. Many distance metrics have been proposed to measure the distance between sets, c.f., [56]. Often the proposed distances are based on non-linear functions (Hausdorff, for instance), which are computationally intensive. Moreover, Hausdorff-type distances can be highly insensitive to the content of the contrasting sets, focusing instead on the boundary cases. Kernels proposed on sets [57] are not also suitable when the set of points is small and therefore, in practice the estimated distribution is inaccurate. In the following we propose a distance based on average pair-wise distances.

Formally, for two sets of points $\mathcal{X}$ and $\mathcal{Y}$, we consider

$$d(\mathcal{X},\mathcal{Y}) = \frac{1}{|\mathcal{X}||\mathcal{Y}|}\sum_{x_i\in\mathcal{X}}\sum_{y_j\in\mathcal{Y}}\|x_i - y_j\|_n, \tag{5.1}$$

where $\|.\|_n$ is a convex norm between two points. It is trivial to show $d(\mathcal{X},\mathcal{Y}) \geq 0$ and $d(\mathcal{X},\mathcal{Y}) = d(\mathcal{Y},\mathcal{X})$. It is also straightforward to prove that (5.1) has the triangular property given the convexity of the norms. Equation (5.1) needs to be slightly modified to have definiteness property (i.e $d(x,y) = 0 \iff x = y$).

$$\mathcal{D}(\mathcal{X},\mathcal{Y}) = \frac{1}{|\mathcal{X} \cup \mathcal{Y}|}\left(\frac{1}{|\mathcal{X}|}\sum_{x_i \in \mathcal{X}}\sum_{y_i \in (\mathcal{Y} \backslash \mathcal{X})}\|x_i - y_j\|_n + \frac{1}{|\mathcal{Y}|}\sum_{x_i \in (\mathcal{X} \backslash \mathcal{Y})}\sum_{y_i \in \mathcal{Y}}\|x_i - y_j\|_n\right). \tag{5.2}$$

Equation (5.2) is symmetric, non-negative and definite due to empty sums in case of equality of $\mathcal{X}$ and $\mathcal{Y}$. To prove that (5.2) has triangular property, one can partition $(\mathcal{D}(\mathcal{X},\mathcal{Y}) + \mathcal{D}(\mathcal{Y},\mathcal{Z}) - \mathcal{D}(\mathcal{X},\mathcal{Z})) \geq 0$ into disjoint sets and observe that given triangular property of (5.1), the required inequality holds for (5.2). Note that in case of $\mathcal{X} \cap \mathcal{Y} = \emptyset$, (5.2) reduces to (5.1). In practice, any sampling is prone to measurement error. This emphasizes the importance of definiteness property imposed by (5.2) even for real-valued signals. We will show in the experimental results that even though the ordering of samples is not preserved within a short segment when modelled as a set, the proposed metric can be used for general purpose alignment. The metric also exhibits invariance to arbitrary temporal permutations. This can be beneficial for non-causal sequences that arise from random delays (e.g., EEG). However, it can also be desirable in video retrieval settings when, for instance, the direction of an activity is not a concern. In the experiments we will demonstrate that this metric is resilient to noise when incorporated into an alignment algorithm. In Section 5.3 we demonstrate how it can be computed efficiently.

5.3 Segmental Pair-HMM (SPHMM)

The Pair HMM, introduced by [14], can be seen as a probabilistic model defined on pairs of sequences (X, Y) that aims to describe their joint likelihood, $P(X, Y|alignment)$. As shown in Figure 5.1, PHMM has three states: M for match, I for insertion and D for deletion. Given two sequences of observations X and Y with n and m samples, respectively, the match state emits a pair of samples (x, y) $x \in X$, $y \in Y$. Insertion and deletion states emit $(x, -)$ and $(-, y)$ respectively where $-$ stands for a gap. This model implements an affine gap penalty which is more general than constant gap penalty typically used in DTW.

In the following we add the notion of segmentation to the pair-HMM formalism. To define the segmentation structure consider a sequence $X = (x_1, x_2, \ldots x_n)$ of length

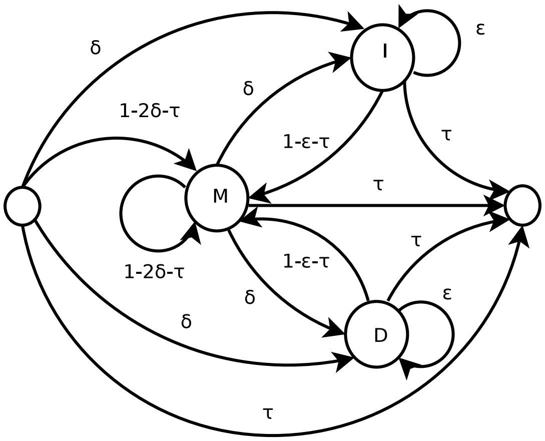


Figure 5.1: Segmental Pair-HMM state-transition diagram

n. A segment $X_{b:e}$, a contiguous subsequence of X, is defined such that $X_{b:e} = (x_b, x_{b+1}, \ldots, x_e)$. Equivalently, the segment is defined by segment indexes $s = (b, b+1, \ldots e)$. We consider non-overlapping and tight segments over X. That is, a complete segmentation of X is defined as $\mathbf{S} = (s_1, s_2, \ldots, s_L)$ such that $b_1 = 1, e_L = n, b_{i+1} = e_i{+}1$. This $\mathbf{S}(X) = (\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_L)$ now defines the segmentation of sequence $X = (x_1 \ldots x_n)$ into segments $((x_1 \ldots x_{e_1}), (x_{b_2} \ldots x_{e_2}) \ldots (x_{b_L} \ldots x_{e_L}))$. Likewise, we define $\mathbf{S}(Y)$ for Y. From this point forward we represent the segmentation of both sequences, X and Y, with $\mathbf{S} = (\mathbf{S}(X), \mathbf{S}(Y)) = ((X_1, X_2, \ldots X_{L_X}), (Y_1, Y_2, \ldots Y_{L_Y}))$.

Given the segments defined by $\mathbf{S}$, a segmental alignment is a sequence of correspondences $Q = (q_1, q_2 \ldots q_T)$ where $q_t = (i_t, j_t), i_t \in \{1, \ldots L_X\}, j_t \in \{1, \ldots L_y\}$ indicating the matching of segments, such that the following monotonic constraints hold:

$$i_t \in \{i_{t-1}, i_{t-1}+1\}, j_t \in \{j_{t-1}, j_{t-1}+1\}. \tag{5.3}$$

The likelihood of one such fixed alignment Q is defined as

$$P(X, Y|\mathbf{S}, Q, \lambda) = \prod_{t=1}^{T} b_{q_t q_{t-1}}(X, Y) \tag{5.4}$$

where λ encompasses the HMM parameters. Here the likelihood of a match is

$$b_{q_t q_{t-1}}(X, Y) = \begin{cases} exp(-\mathcal{D}(\mathrm{X}_{i_t}, \mathrm{Y}_{j_t})) \cdot \Psi(|\mathrm{X}_{i_t}|, |\mathrm{Y}_{j_t}|) & i_t = i_{t-1}+1, \\ & j_t = j_{t-1}+1 \\ exp(-\sigma_g |\mathrm{X}_{i_t}|) & i_t = i_{t-1}+1, j_t = j_{t-1} \\ exp(-\sigma_g |\mathrm{Y}_{j_t}|) & i_t = i_{t-1}, j_t = j_{t-1}+1 \end{cases} \tag{5.5}$$

where $\mathcal{D}(\mathrm{X}_{i_t}, \mathrm{Y}_{j_t})$ is the distance between two segments, defined in (5.2), Ψ specifies the distribution of the corresponding segment lengths, and σ_g is a scaling factor. The transition probabilities in the match sequence are defined by the state transition graph in Figure 5.1 and are denoted by a. For instance,

$$a_{q_t q_{t-1} q_{t-2}} = \begin{cases} \delta & , \quad i_{t-1} = i_{t-2} + 1, i_t = i_{t-1}, \\ & \quad\; j_{t-1} = j_{t-2} + 1, j_t = j_{t-1} + 1 \\ \epsilon & , \quad i_{t-1} = i_{t-2} + 1, i_t = i_{t-1} + 1, \\ & \quad\; j_{t-1} = j_{t-2}, j_t = j_{t-1} \\ \tau & , \quad i_{t-1} = i_{t-2} + 1, i_t = T, \\ & \quad\; j_{t-1} = j_{t-2} + 1, j_t = T \\ \text{etc.} & \end{cases} \tag{5.6}$$

with initial transitions, e.g.,

$$a^{(0)}_{q_1} = \begin{cases} \delta & , \quad i_1 = 0, j_1 = 1, \quad \text{or} \quad i_1 = 1, j_1 = 0 \\ 1 - 2\delta - \tau & , \quad i_1 = 1, j_1 = 1 \\ \tau & , \quad i_1 = 0, j_1 = 0 \end{cases} \tag{5.7}$$

where $i_1 = 0$ stands for deleting the first segment of X and similarly $j_1 = 0$ denotes deleting the first segment of Y. Ψ in (5.5) can be learned from the data or given as a prior distribution, e.g., uniform. Note that the first case of (5.5) defines the observation probability of matching two segments (associated with state M in Fig 5.1) while other cases correspond to gap operations (states I and D).

5.3.1 Inference in SPHMM

An optimal alignment for a fixed segmentation $\mathbf{S}$ can be found as

$$Q^* = \arg\max_Q P(Q|X, Y, \mathbf{S}, \lambda) = \arg\max_Q P(X, Y|Q, \mathbf{S}, \lambda)P(Q). \tag{5.8}$$

The prior on Q in (5.8) can be uniform or can encode traditional band-priors such as the Sakoe-Chiba band. (5.4)-(5.8) show that the optimal alignment is the Viterbi path for observing segmented sequences (X, Y).

It is possible to find an optimal segmentation $\mathbf{S}^*$, together with the optimal alignment, as

$$Q^*, \mathbf{S}^* = \arg\max_{Q,\mathbf{S}} P(\mathbf{S}, Q|X, Y, \lambda) = \arg\max_{Q,\mathbf{S}} P(X, Y|\mathbf{S}, Q, \lambda)P(\mathbf{S})P(Q). \tag{5.9}$$

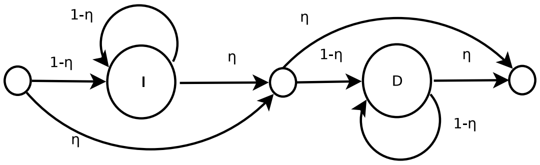


Figure 5.2: Pair-HMM null model.

We specify uniform prior on $\mathbf{S}$. To assert that the alignment likelihood indicates a relationship between the contrasting sequences rather than a random match, one needs to compare this likelihood to that of a null model. This null model deletes all segments of one sequence and inserts segments of the contrasting sequence 5.2. Therefore, the likelihood of the null model is

$$P(X,Y|\mathbf{S},R) = \left(\eta(1-\eta)^{L_X}\prod_{i=1}^{L_X} exp(-\sigma_g|X_i|)\right)\left(\eta(1-\eta)^{L_Y}\prod_{j=1}^{L_Y} exp(-\sigma_g|Y_i|)\right) \tag{5.10}$$

where R is the null HMM model with transitions depicted in Figure 5.2 and observation model similar to 5.5 (except for the the first equation which is the likelihood of observing a match between two segments). Thus, we intend to evaluate

$$Q^*, \mathbf{S}^* = \arg\max_{Q,\mathbf{S}} \frac{P(X,Y|\mathbf{S},Q,\lambda)P(Q)}{P(X,Y|\mathbf{S},R)}. \tag{5.11}$$

Note that the prior on segmentation is cancelled out. It is possible to evaluate both SPHMM and null model in a single pass over the sequences. In particular, one can assign every match in the SPHMM model to a pair of insertion and a deletion and likewise assign every gap operation to its corresponding insertion or deletion in the null model. Thus, it would be straightforward to formulate reward for match and penalties for opening and extending a gap by expanding (5.11) with respect to (5.4) and (5.10). This helps with seeing this formulation in the context of an alignment dynamic algorithm with affine gap penalty. In particular, for two segments X_i and Y_j the matching reward is

$$r_{mm}(X_i,Y_j) = \frac{1-2\delta-\tau}{(1-\eta)^2} \tag{5.12}$$

for staying in match state or

$$r_{gm}(X_i,Y_j) = \frac{1-\epsilon-\tau}{(1-\eta)^2} \tag{5.13}$$

for transitioning from a gap state to match. Consequently, the gap opening penalty for X_i is

$$r_{op}(X_i) = \frac{\delta}{(1-\eta)} \tag{5.14}$$

and gap extension penalty is

$$r_{ex}(X_i) = \frac{\epsilon}{(1-\eta)}. \tag{5.15}$$

By transferring into log-odds ration the relationship between a Viterbi algorithm and a dynamic programming for alignment is evident. The resulting algorithm is a straightforward extension of the best-path algorithm described in [14] to segmental model by searching over all permissible segment lengths at each step of the recursion considering the match rewards a gap penalties in (5.12)-(5.15). That is, in every state, all possible segments are considered and the segmentation that leads to the highest ratio of posteriors (5.11) is chosen. To make this procedure computationally tractable one may impose a maximum constraint on the segment length.

Complexity: The time complexity of (5.11) is dependent both on the lengths of segments in each sequence and the length of the sequences themselves. Given that the number of states is fixed and small, one can prove that the time complexity of the dynamic programming (or marginal matching) algorithm is $O(l_X l_Y mn)$ where l_X and l_Y are the maximum segment lengths and n and m are the lengths of sequences X and Y, respectively. To compute the distance between two segments, one can employ the summed area table technique [58] to improve the performance. That is, the pairwise distances of all pairs of samples are pre-calculated and the summed area table is constructed. Then within the matching procedure only a few additions are required to compute the distance. With a simple memorization technique the complexity can be decreased to $O(\max(l_x, l_y)mn)$ .Usually, l_X and l_Y are not too long relative to the sequence lengths. Thus, the overall time complexity is typically a small constant factor away from that of the regular DTW.

5.3.2 Marginal matching likelihood

This subsection introduces an approximation to forward algorithm for segmental pair-HMM. Let us define Γ to be the set of all possible segmentations of two sequences X and Y with m and n samples, respectively. Also assume that Π is the set of all segmental alignments between X and Y. Using forward algorithm one can estimate the following

$$P(X,Y|\lambda) = \sum_{\mathbf{S}\in\Gamma}\sum_{Q\in\Pi} P(X,Y|Q,\mathbf{S},\lambda)P(\mathbf{S})P(Q). \tag{5.16}$$

We will assume $P(\mathbf{S})$ to be uniform. Computing (5.16) is not tractable for every possible segmentation. Therefore, we approximate the joint probability of X and Y by explicitly marginalizing over all alignments. That is, we approximate (5.16) by estimating $P(X,Y|S^*)$ at each step where S^* is a partially optimal segmentation. Specifically, S^* denotes the segments that are optimal only for a partial alignment of the sequences X and Y up to the current step of the algorithm. We use the following recursion to define this approximation.

$$\begin{aligned} P\left(X_{1:i},Y_{1:j}|q_tq_{t-1},\left(S^*(X_{1:(i-k)}),S^*(Y_{1:(j-l)})\right)\lambda\right) &= b_{q_tq_{t-1}}\times \\ \max_{\mathbf{S}'\in\binom{\Gamma(X_{1:(i-k)}),}{\Gamma(y_{1:(j-l)})}} \sum_{\substack{Q'\in\\ \Pi_{(i-k),(j-l)}}} & P\left(X_{1:(i-k)},Y_{1:(j-l)}|Q',\lambda,\mathbf{S}'\right) \end{aligned} \tag{5.17}$$

where

$$\left(S^*\left(X_{1:i}\right),S^*\left(Y_{1:j}\right)\right) = \arg\max_{\mathbf{S}'\in(\Gamma(X_{1:i}),\Gamma(Y_{1:j}))} \sum_{Q'\in\Pi_{i,j}} P(X_{1:i},Y_{1:j}|Q',\lambda,\mathbf{S}'). \tag{5.18}$$

In (5.17) and (5.18) k and l are permissible segment lengths for X and Y. $\Gamma(.)$ is the set of all segmentations while $S^*(.)$ denotes the approximated segmentation of the given input sequence. $\Pi_{i,j}$ is the set of all possible alignments of X and Y up to x_i and y_j. In (5.17) q_tq_{t-1} defines the current state the same way we defined it in (5.5). The second term of right hand side of (5.17) finds the maximum marginalized likelihood over aligning partial sequences given all possible segmentations up to x_{i-k}, y_{j-l}. The result of applying this recursive algorithm is the approximated marginalized likelihood of X and Y. This is useful in classification problems where one is not necessarily interested in alignment path or optimal segmentation but a reliable likelihood is more desirable. In this paper however, we mainly show the result of the dynamic programming algorithm that arises from (5.11). The dynamic programming algorithm not only provides us with a likelihood that later can be used as a measure of similarity, but also yields the optimal alignment path and segmentation which is essential to our analysis. We observed superior classification accuracy using the marginal matching algorithm in EEG classification (Section 5.6).

5.3.3 Learning SPHMM parameters

Algorithm 1 Learning algorithm for SPHMM.$\#(A \to B)$ denotes the number of transitions from state A to state B decoded by the Viterbi algorithm.

```
  Initialization
  Randomly initialize $\delta, \epsilon$ and $\tau$. Set $\Psi(i,j)$ to uniform.
  repeat
    E-step: Align training sequences using the Viterbi algorithm described in Section 5.3
    M-step:
      1. Re-estimate transition parameters: $\delta = \frac{\#(M \to I)+\#(M \to D)}{2\#(M \to *)}$, $\epsilon = \frac{\#(I \to I)+\#(D \to D)}{\#(I \to *)+\#(D \to *)}$ and $\tau = 1 - 2\delta - \epsilon$.
      2. Re-estimate segment length distribution, $\Psi(i,j) = \frac{\#(|\mathrm{X}_{t_X}|=i,|\mathrm{Y}_{t_Y}|=j)}{\#segments}$ $\forall t_\in \{1 \ldots L_X\}, t_Y \in \{1 \ldots L_Y\}$.
      3. Tune the parameters using (5.22) with ($\delta$,$\epsilon$ and $\tau$) as the initial values (project back if needed to respect the feasibility of the starting point)
  until Convergence.
```

To learn the parameters of SPHMM one can use a standard expectation maximization algorithm typically used to train HMM parameters [59]. The parameter of the null model cannot be trained using EM algorithm and must remain constant during training in order to have the consistent reference model. One good choice to set η is the maximum likelihood estimate of (5.10). That is,

$$\eta = \frac{2}{L_X + L_Y + 2} \tag{5.19}$$

where L_X and L_Y are number of segments (based on the prior) in each sequence. In our experiments we noticed choosing η according to (5.19) may result into overfitting to the training set in a classification problem and therefore suggest choosing $\eta > .5$ in that case.

The standard EM algorithm, does not respect certain constraints that must hold when one designs an alignment algorithm. Those constrains are designed to keep matching reward and gap penalties (Eq. (5.13)-5.15) within certain bounds. In particular one would like to have

$$1 < r_{mm}, r_{gm} < z_m, \tag{5.20}$$

$$z_g < r_{op}, r_{ex} < 1, \tag{5.21}$$

where $1 < z_m$ and $0 < z_g < 1$ are real numbers. In our experiments we have set $z_m = exp(5)$ and $z_g = exp(-10)$ which provide a reasonable range for learning the parameters. Maximizing the contribution of matching rewards and gap penalties while satisfying above constraints will lead to solving

$$(\delta^*, \epsilon^*, \tau^*) = \arg\max_{\delta,\epsilon,\tau} \left(\hat{c}_{mm} \log\left(1 - 2\delta - \tau\right) + \hat{c}_{gm} \log(1 - \epsilon - \tau) + \hat{c}_{op} \log(\delta) + \hat{c}_{ex} \log(\epsilon)\right) \tag{5.22}$$

st.

$$2\log(1-\eta) < \log\left(1 - 2\delta - \tau\right) < \log(z_m) + 2\log(1-\eta) \tag{5.23}$$

$$2\log(1-\eta) < \log\left(1 - \epsilon - \tau\right) < \log(z_m) + 2\log(1-\eta) \tag{5.24}$$

$$\log(z_g) + log(1-\eta) < \log(\delta), \log(\epsilon) < \log(1-\eta) \tag{5.25}$$

$$\log(\tau) < 0 \tag{5.26}$$

where for N alignments in the training set

$$\hat{c}_{mm} = \frac{\#(M \to M)}{N} \tag{5.27}$$

$$\hat{c}_{gm} = \frac{\#((I \ or \ D) \to M)}{N} \tag{5.28}$$

$$\hat{c}_{op} = \frac{\#(M \to (I \ or \ D))}{N} \tag{5.29}$$

$$\hat{c}_{ex} = \frac{\#(I \to I) + \#(D \to D)}{N} \tag{5.30}$$

where $\#(A \to B)$ stands for the number of transitions from state A to B. In (5.22), we have transferred to log-space for numerical stability and used the fact that parameter of the null model (η) will not be updated. One can transfer (5.22) into a linear programming by adding $\log(\tau)$ to the objective function and effectively maximize the likelihood of the average Markov model (transitions) under mentioned constraints.

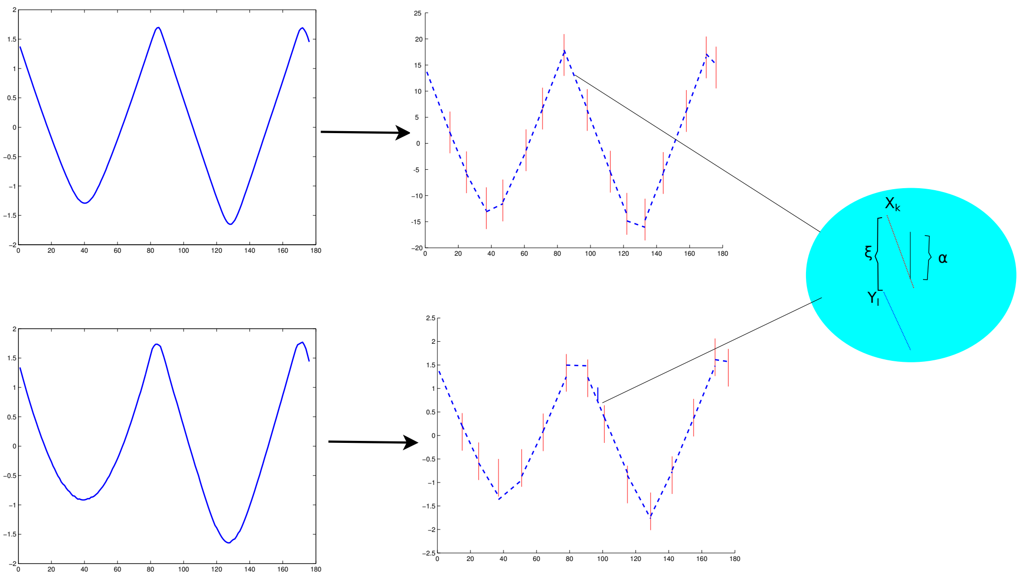


Figure 5.3: Piecewise linear approximation of a sequence based on fixed segments. The right plot shows the segments and the approximated lines (dashed lines). Two of the segments that are to be matched are magnified.

Finally, one can consider the algorithm in Alg.1 for learning the parameters of SPHMM. Note that the inference step is approximated with the dynamic programming resulted from (5.11). One can incorporate the method described in Section 5.3.2 to approximate the forward algorithm and use it in a forward-backward learning task (backward algorithm can also be approximated similarly) for estimating the posterior and finally learn the parameters including the distribution of segment lengths.

5.4 Discussion on Segment Size and Noise Suppression

Consider two sequences, X and Y, that are to be sent through a noisy channel. In the source, both sequences are segmented and each segment is approximated by a line then the obtained lines are re-sampled and transmitted through the channel. To observe the mechanism of noise suppression based on the proposed distance in Section 5.2, we consider aligning of the two signals in the destination while an impulse noise is added to one of the sequences during transmission due to some interference. Formally, let X_k and Y_l be two of the line segments starting from the same time index. That is, $x_{b_k+i} = \beta(b_k + i) + \xi$, $y_{b_l+j} = \beta(b_l + j)+$, where $b_k = b_l$. Suppose an impulse corrupts X_k at $i = j == t_c$ $(1 \leq t_c \leq \min(|\mathrm{X}_k|, |\mathrm{Y}_l|))$ such that $x_{b_k+t_c} = y_{b_l+t_c} + \xi + \alpha$

(Figure5.3) Assuming $\mathrm{X}_k \cap \mathrm{Y}_l = \emptyset$ the distance of two segments will be smaller than a point-to-point match only if the following inequality holds

$$\begin{aligned}\mathcal{D}(\mathrm{X}_k, \mathrm{Y}_l) &= \frac{1}{|\mathrm{X}_k||\mathrm{Y}_l|} \sum_{i=1}^{|\mathrm{X}_k|} \sum_{j=1}^{|\mathrm{Y}_l|} \|x_{b_k+i} - y_{b_l+j}\| \\ &\leq \frac{|\beta|}{|\mathrm{X}_k||\mathrm{Y}_l|} \left(\sum_{i=1}^{|\mathrm{X}_k|} \sum_{j=1}^{|\mathrm{Y}_l|} \|y_{b_l+i} - y_{b_l+j}\| \right) + \frac{(|\mathrm{X}_k| - 1)}{|\mathrm{X}_k|}|\xi| + \frac{1}{|\mathrm{X}_k|}|\xi + \alpha| < |\xi + \alpha|.\end{aligned} \tag{5.31}$$

Note that since $\mathrm{X}_k \cap \mathrm{Y}_l = \emptyset$, the original distance described by (5.2) is reduced to (5.1). We used the convexity of the norm in the above. Therefore,

$$\sum_{i=1}^{|\mathrm{X}_k|} \sum_{j=1}^{|\mathrm{Y}_l|} \|y_{b_l+i} - y_{b_l+j}\| < \frac{|\mathrm{Y}_l|(|\mathrm{X}_k| - 1)\left[|\alpha + \xi| - |\xi|\right]}{|\beta|} \tag{5.32}$$

has to hold. One can observe that as long as $\alpha < -|\xi| - \xi$ or $\alpha > |\xi| - \xi$, by increasing $|\mathrm{X}_k|$ (or $|\mathrm{Y}_l|$) while the slope of the line (β) is kept constant, the left hand side of (5.32) grows quadratically while the right hand side grows linearly which leads to bounded segment length. Furthermore, if $|\beta| \to 0$ then as long as $|\mathrm{X}_k| > 1$, (5.32) is a tautology meaning that longer segment length is always favourable. Consequently, As β increases a point-to-point match becomes more likely. The result of such distance metric is that it flattens the signal around an impulse not only according to its neighbourhood but also to the contrasting sequence. This leads to a dynamic noise removal. Therefore, if the impulse is in fact a characteristic of the signal and not a noise, it will not be removed (similar to DTW) but in case of noisy impulse, it will be averaged and flattened.

5.5 Segmental Matching

In our experiments we observed that during learning SPHMM, the probability of transitioning from match state to gap states can be decreased substantially without significantly affecting the likelihood or alignment path. Given this observation, it is reasonable to expect a single match operation coupled with adaptive segmentation be able to approximate the alignment. Let $\Gamma_m \subset \Gamma$ be the collection of all possible segmentation of X and Y such that: 1) the number of segments is equal in each segmentation, $L = L_X = L_Y$; 2) Corresponding segments are then matched, i.e. the alignment path $Q = (q_1, q_2, \dots q_L)$ where $q_i = (i, i)$. In other words, the alignment is recovered through segmentation. That is,

$$P(X, Y) = \sum_{\mathbf{S} \in \Gamma_m} P(X, Y|\mathbf{S}) P(\mathbf{S}) \tag{5.33}$$

where

$$P(X,Y|\mathbf{S}) = \prod_{t=1}^{L} \exp\left(-\frac{1}{\sigma}D(X_t,Y_t)\right)\Psi(|X_t|,|Y_t|) \tag{5.34}$$

which is the likelihood of matching two segments in the original SPHMM model. $D(\cdot,\cdot)$ can be any distance metric on sets. Therefore, the joint likelihood of X and Y is maximized by searching over all possible segmentation. That is,

$$P^*(X,Y) = \max_{\mathbf{S}\in\Gamma_m} P(X,Y|\mathbf{S})P(\mathbf{S}) \tag{5.35}$$

and consequently one may obtain the optimal segmentation as

$$\mathbf{S}^* = \arg\max_{\mathbf{S}\in\Gamma_m} P(X,Y|\mathbf{S})P(\mathbf{S}) \tag{5.36}$$

where we assume uniform prior on segmentation. A non-uniform prior on segmentation can result into different alignments by favouring longer or shorter segments on different intervals of the sequences. It is possible to compare this model with a random model similar to (5.10). In that case the prior on segmentation will again cancel out and each matching will be compared to a pair of deletion and insertion.

Removing the two gap operations not only reduces the computational effort incurred by joint segmentation and alignment but also enables one to use bounding methods for particular representations of time-series to further prune the unnecessary computation and speedup the matching. For instance, if the time-series can be locally represented using Bag-of-Words and histogram, often found as a representation in documents or complex video signals, Lampert et al [60] have designed bounds on the distance between two segments given a minimum and maximum segment length and their corresponding histograms. We leverage this fact to reduce the computational time of the method proposed in this Section.

5.5.1 Bounding Histogram Distances

Bag-of-Words (BoW): is a popular representation that has been successfully used by researchers [61, 62]. In this representation extracted features are clustered into several codewords using a clustering method such as k-means. Similar features described by the same codeword are then counted together and form a histogram for a single or a collection of frames. Therefore, given a histogram map $\phi_{b_i:e_i}(.)$, we denote an H-bin histogram of a contiguous segment $b_i : e_i = (b_i, b_i+1, \ldots, e_i-1, e_i)$ as $X_{b_i:e_i} = \phi_{b_i:e_i}(V)$ or X_i for short.

Given the maximum segment length l_{max}, the minimum segment length l_{min}, and two segments of sequence X and Y, starting from b_i and b_j, respectively, we denote the maximum length segments by $\overline{X}_{b_i} = X_{b_i:b_i+l_{max}}$ and $\overline{Y}_{b_j} = Y_{b_j:b_j+l_{max}}$. Likewise, the minimum length segments are denoted by $\underline{X}_{b_i} = X_{b_i:b_i+l_{min}}$ and $\underline{Y}_{b_j} = Y_{b_j:b_j+l_{min}}$. We are aiming to bound the distance between the histogram features of any possible segment starting from X_{b_i} extending to $X_{b_i+l_{max}}$and Y_{b_j} extending maximally to $Y_{b_i+l_{max}}$. Note that even though we use the same l_{min} and l_{max} for both sequences, it is not a requirement of our method and is used only to simplify the notation. The bin counts of X_{b_i} and Y_{b_j} are bounded as

$$\underline{X}^h_{b_i} \leq X^h_{b_i:b_i+k} \leq \overline{X}^h_{b_i}, (l_{min} \leq k \leq l_{max}) \tag{5.37}$$

$$\underline{Y}^h_{b_j} \leq Y^h_{b_j:b_j+z} \leq \overline{Y}^h_{b_j}, (l_{min} \leq z \leq l_{max}) \tag{5.38}$$

where $X^h_{.}$ and $Y^h_{.}$ denote the histogram bin h.

One can easily extend (5.37, 5.38) to normalized histogram noting that $|\underline{X}_{b_i}| \leq X_{b_i:b_i+k} \leq |\overline{X}_{b_i}|$. That is,

$$\frac{\underline{X}^h_{b_i}}{|\overline{X}_{b_i}|} \leq \hat{X}^h_{b_i:b_i+k} \leq \frac{\overline{X}^h_{b_i}}{|\underline{X}_{b_i}|}, (l_{min} \leq k \leq l_{max}) \tag{5.39}$$

$$\frac{\underline{Y}^h_{b_j}}{|\overline{Y}_{b_j}|} \leq \hat{Y}^h_{b_j:b_j+z} \leq \frac{\overline{Y}^h_{b_j}}{|\underline{Y}_{b_i}|}, (l_{min} \leq z \leq l_{max}) \tag{5.40}$$

It is straightforward to observe

$$\min(\underline{X}^h_{b_i}, \underline{Y}^h_{b_j}) \leq \min(X^h_{b_i:b_i+k}, Y^h_{b_j:b_j+z}) \leq \min(\overline{X}^h_{b_i}, \overline{Y}^h_{b_j}) \tag{5.41}$$

$$\max(\underline{X}^h_{b_i}, \underline{Y}^h_{b_j}) \leq \max(X^h_{b_i:b_i+k}, Y^h_{b_j:b_j+z}) \leq \max(\overline{X}^h_{b_i}, \overline{Y}^h_{b_j}) \tag{5.42}$$

for $l_{min} \leq k, z \leq l_{max}$. Following [62] one may construct the bounds on popular histogram distances. For completeness of presentation these bounds are included below.

Bounding l_1 distance: Noting that $|a-b| = \max(a,b) - \min(a,b)$ and a simple reordering of (5.41, 5.42) one can observe that

$$\begin{aligned}\max(\underline{X}^h_{b_i}, \underline{Y}^h_{b_j}) - \min(\overline{X}^h_{b_i}, \overline{Y}^h_{b_j}) \leq |X^h_{b_i:b_i+k} - Y^h_{b_j:b_j+z}| \leq \\ \max(\overline{X}^h_{b_i}, \overline{Y}^h_{b_j}) - \min(\underline{X}^h_{b_i}, \underline{Y}^h_{b_j})\end{aligned} \tag{5.43}$$

for $l_{min} \leq k, z \leq l_{max}$. The bounds on l_1 distance are then the summation over all bins. That is,

$$l^{l_1}_b(X_{b_i}, Y_{b_j}, m, l) = \sum_{h=1}^{H} \max(\underline{X}^h_{b_i}, \underline{Y}^h_{b_j}) - \min(\overline{X}^h_{b_i}, \overline{Y}^h_{b_j}) \tag{5.44}$$

$$u^{l_1}_b(X_{b_i}, Y_{b_j}, m, l) = \sum_{h=1}^{H} \max(\overline{X}^h_{b_i}, \overline{Y}^h_{b_j}) - \min(\underline{X}^h_{b_i}, \underline{Y}^h_{b_j}) \tag{5.45}$$

and for normalized histograms

$$\hat{l}_b^{l_1}(X_{b_i}, Y_{b_j}, l_{min}, l_{max}) = \sum_{h=1}^{H} \left(\max\left(\frac{\underline{X}_{b_i}^h}{|\overline{X}_{b_i}^h|}, \frac{\underline{Y}_{b_j}^h}{|\overline{Y}_{b_j}^h|} \right) - \min\left(\frac{\overline{X}_{b_i}^h}{|\underline{X}_{b_i}^h|}, \frac{\overline{Y}_{b_j}^h}{|\underline{Y}_{b_j}^h|} \right) \right) \tag{5.46}$$

$$\hat{u}_b^{l_1}(X_{b_i}, Y_{b_j}, l_{min}, l_{max}) = \sum_{h=1}^{H} \left(\max\left(\frac{\overline{X}_{b_i}^h}{|\underline{X}_{b_i}^h|}, \frac{\overline{Y}_{b_j}^h}{|\underline{Y}_{b_j}^h|} \right) - \min\left(\frac{\overline{X}_{b_i}^h}{|\underline{X}_{b_i}^h|}, \frac{\overline{Y}_{b_j}^h}{|\underline{Y}_{b_j}^h|} \right) \right). \tag{5.47}$$

Histogram intersection and χ^2 distances can also be derived in the same way.

Bounding histogram intersection distance: Histogram intersection distance is defined as

$$d_{\cap}(\phi_X^H, \phi_Y^H) = -\sum_{h=1}^{H} \min(\hat{X}^h, \hat{Y}^h) \tag{5.48}$$

using (5.39), (5.40) the corresponding lower and upper bound is

$$\hat{l}_b^{\cap}(X_{b_i}, Y_{b_j}, l_{min}, l_{max}) = -\sum_{h=1}^{H} \min\left(\frac{\overline{X}_{b_i}^h}{|\underline{X}_{b_i}^h|}, \frac{\overline{Y}_{b_j}^h}{|\underline{Y}_{b_j}^h|} \right) \tag{5.49}$$

$$\hat{u}_b^{\cap}(X_{b_i}, Y_{b_j}, l_{min}, l_{max}) = -\sum_{h=1}^{H} \min\left(\frac{\underline{X}_{b_i}^h}{|\overline{X}_{b_i}^h|}, \frac{\underline{Y}_{b_j}^h}{|\overline{Y}_{b_j}^h|} \right) \tag{5.50}$$

Bounding χ^2 distance: χ^2 distance is defined as

$$d_{\chi^2}(\phi_X^H, \phi_Y^H) = \sum_{h=1}^{H} \frac{\left(\hat{X}^h - \hat{Y}^h\right)^2}{\hat{X}^h + \hat{Y}^h}. \tag{5.51}$$

Using the normalized bounds on l_1 distance i.e. (5.46) and (5.47) one can easily prove

$$\hat{l}_b^{\chi^2}(X_{b_i}, Y_{b_j}, l_{min}, l_{max}) = \sum_{h=1}^{H} \frac{\left(\max(0, \hat{l}_b^{l_1})\right)^2}{\frac{\overline{X}_{b_i}^h}{|\underline{X}_{b_i}^h|} + \frac{\overline{Y}_{b_j}^h}{|\underline{Y}_{b_i}^h|}} \tag{5.52}$$

$$\hat{u}_b^{\chi^2}(X_{b_i}, Y_{b_j}, l_{min}, l_{max}) = \sum_{h=1}^{H} \frac{(\hat{u}_b^{l_1})^2}{\frac{\underline{X}_{b_i}^h}{|\overline{X}_{b_i}^h|} + \frac{\underline{Y}_{b_j}^h}{|\overline{Y}_{b_j}^h|}} \tag{5.53}$$

5.5.2 Fast Segmental Matching (Fast-SM)

We propose a recursive algorithm that starts matching from the end of the contrasting sequences. Each segmental match is effectively finding the joint likelihood of X_i and Y_i. Within each match we search over all possible segmentations up to the maximum segment length. That is, given l_{max} and l_{min}, for $i = L, \ldots 1$, $j = L, \ldots 1$ and considering uniform prior on segments the likelihood of matching is

$$P(X_{b_i}, Y_{b_j}) = \max_{l_{min} \leq k, z \leq l_{max}} \exp(-D(X_{b_i - k:i}, Y_{b_j - z:j}))P(X_{b_i - k - 1}, Y_{b_j - z - 1}). \tag{5.54}$$

In other words, (5.54) is the optimal maximum likelihood of matching segments by searching over the likelihood of the last pair of segments in both sequences and all possible segmentation starting from the current point.

We assume that the likelihood of correspondences in the local neighbourhood is approximately constant. Therefore, before executing a recursion we examine its approximated likelihood against the best one found so far. We define P^* as the maximal likelihood calculated for the alignment up to the preceding segment to $(X_{b_i-k-1}, Y_{b_j-z-1})$, we have

$$P^* = \max_{\substack{l_{min} \leq k' < k \\ l_{min} \leq z' < z}} \left\{ P(X_{b_i-k'-1}, Y_{b_j-z'-1}) \cdot \exp(D(X^*_{:b_i-k'-1}, Y^*_{:b_j-z'-1})) \right\} \tag{5.55}$$

where $X^*_{:b_i-k'-1}$ and $Y^*_{:b_j-z'-1}$ denote the best segments extended up to $b_i - k' - 1$ and $b_j - z' - 1$, respectively. Note that all elements required to compute P^* are already calculated and no extra effort is needed to determine it. The bounding is then defined as

$$\tilde{P}(X_{b_i-k-1}, Y_{b_j-z-1}) \leq P^* \exp(-l_b(X_{:b_i-k-1}, Y_{:b_j-z-1}, l_{min}, l_{max})) \tag{5.56}$$

where l_b is the corresponding lower bound defined in subsection 5.5.1. The idea is illustrated in Figure 5.4. That is, we propose to bound the likelihood of a segment by the the product of the maximal likelihood in its neighbourhood and the upper bound on the likelihood of matching any two segments extended within its boundaries.Therefore, using (5.56) one can obtain an approximated upper bound on $P(X_{b_i-k-1}, Y_{b_j-z-1})$ and compare it against the best likelihood obtained for the previous segment. We use the term "*approximated upper bound*" since we have made the assumption of smoothness on the local likelihood. If $\tilde{P}(X_{b_i-k-1}, Y_{j-z-1})$ is lower than the best likelihood for the preceding segment obtained so far, we do not expand the recursion and set that corresponding likelihood to its minimum by

$$P(X_{b_i-k-1}, Y_{b_j-z-1}) = P^* \exp(-u_b(X_{:b_i-k-1}, Y_{b_j-z-1}, l_{min}, l_{max})). \tag{5.57}$$

By setting $P(X_{b_i-k-1}, Y_{b_j-z-1})$ to the minimum likelihood we avoid further expansion of this path even if this point is visited again during the segmentation.

Another technique that contributes to improving the computational performance of our approach stems from the BOW representation. This representation allows one to use the idea of *integral image* [63] to calculate the cumulative sum of the histograms and thus obtain the required segment using a single subtraction operation. That is, if I is a sequence of such cumulative sums one can obtain a segment from b_i to e_i simply by $X_{b_i:e_i} = I_{e_i} - I_{b_i-1}$.

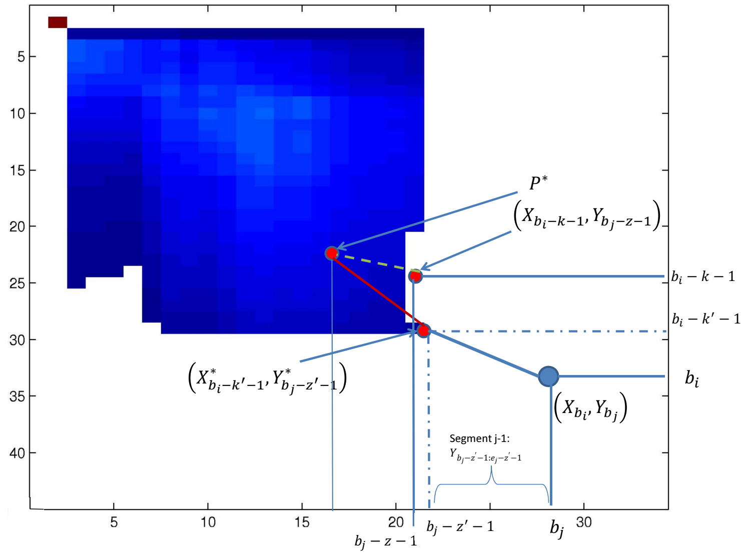


Figure 5.4: Approximate bounding of the likelihood. Axes show the index (time) of contrasting sequences. The shaded area shows the highest alignment likelihood for each correspondence given its optimal segmentation inferred so far. At segment (X_{b_i}, Y_{b_j}) we are verifying whether we should expand the new segment to $(X_{b_i-k-1}, Y_{b_j-z-1})$. The best likelihood is achieved by connecting to segment $(X_{b_i} - k' - 1, Y_{b_j} - z' - 1)$ where $l_{min} \leq k' < k$ and $l_{min} \leq z' < z$. Therefore, we can find P^* from which is the likelihood of segmentation up to the end of $(X_{b_i} - k' - 1, Y_{b_j} - z' - 1)$. Then we assume the smoothness on the neighbouring likelihood around that point and extend a hypothetical segment from$(X_{b_i-k-1}, Y_{b_j-z-1})$ to it which can be bounded.

5.6 Experimental Results

We intend to use the likelihood reported by the dynamic programming algorithm (or marginal matching likelihood) that arises from (5.11) for each alignment as the similarity measure for classification. This is a common way for asserting the goodness of an alignment algorithm quantitatively [21]. Note that the null model is the same for all sequences within a dataset. We first apply SPHMM on synthetic data to qualitatively assess its performance and also demonstrate its capability in aligning sequences generated by non-causal processes. We then examine our proposed approach on the benchmark data, the first dataset (data1) from the UC Riverside "time-series classification page" [64]. To show that our method is able to deal with non-causal and noisy real-world time-series we also apply it to a publicly available EEG data set. Finally, we show that SPHMM can improve the performance of activity classification on a subset of HDM05 MoCap data. Segmental matching (SM) and fast segmental matching (Fast-SM) are applied to an activity recognition problem on a publicly available dataset and their superior performance compared to other algorithms in the literature is demonstrated.

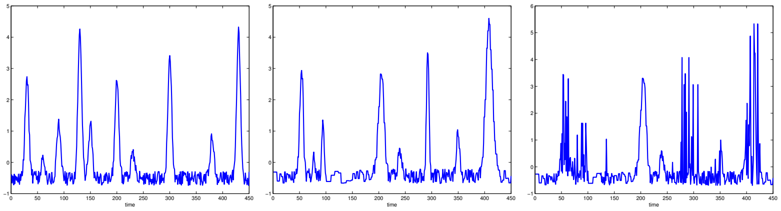
Euclidean distance is used as the measure of distance between two samples. We observed that L_1 norm can slightly, but not significantly, improve the results in case of excessive noise but we do not include those results. Referring to our discussion in Sec 5.2, employing other distance metrics between sets (such as Hausdorff) resulted in significantly inferior performance especially in noisy data and rendered the alignment of long sequences computationally intractable. Therefore, those results are also omitted from the manuscript. Throughout this section l_X and l_Y denote the maximum allowed lengths of the segments. We have also assumed the scaling parameter of gap operations (Equation (5.5)) to be $\sigma_g = 1$. In all experiments the classifier is the baseline 1-Nearest Neighbour (1-NN). We have exclusively used 1-NN to shift the attention from the classifier design to the similarity measure.

5.6.1 Synthetic Data I

To demonstrate that our proposed approach can handle non-causal sequences and also have a qualitative comparison with DTW we generated a synthetic dataset and designed the following experiment. 100 sequences are generated from the model

$$T_j(t) = \sum_{i=1}^{10} (\pi_i + \nu_t) \exp\left((t-\mu)^2\right) + \omega_t. \tag{5.58}$$

The time length of all sequences is 450. Peaks in the sequences occur at mean times $\mu = [30, 60, 90, 130, 150, 200, 230, 300, 380, 430]$. The weights are set to $\pi = [7, 1, 3, 10, 3, 6, 1, 8, 3, 10]$and



(a) Original

(b) Warped

(c) Non-causal and warped

Figure 5.5: An instance of a generated sequence and its corresponding warped sequence and its non-causal version.

are corrupted by white independent noise. $\omega_t, \nu_t = N(0,1)$. We use a monotonic function for the alignment ground truth such that

$$f(t) = \begin{cases} 1 + 0.01 \cdot t^2 & t \leq 100 \\ 310 + 150 \cdot tanh(t/100) & t > 100. \end{cases} \tag{5.59}$$

To introduce non-causality we add noise to (5.59) within four intervals such that

$$f^n(t) = \begin{cases} f(t) + N(0,10) & B_i \leq t \leq E_i \quad \forall i \\ f(t) & otherwise. \end{cases} \tag{5.60}$$

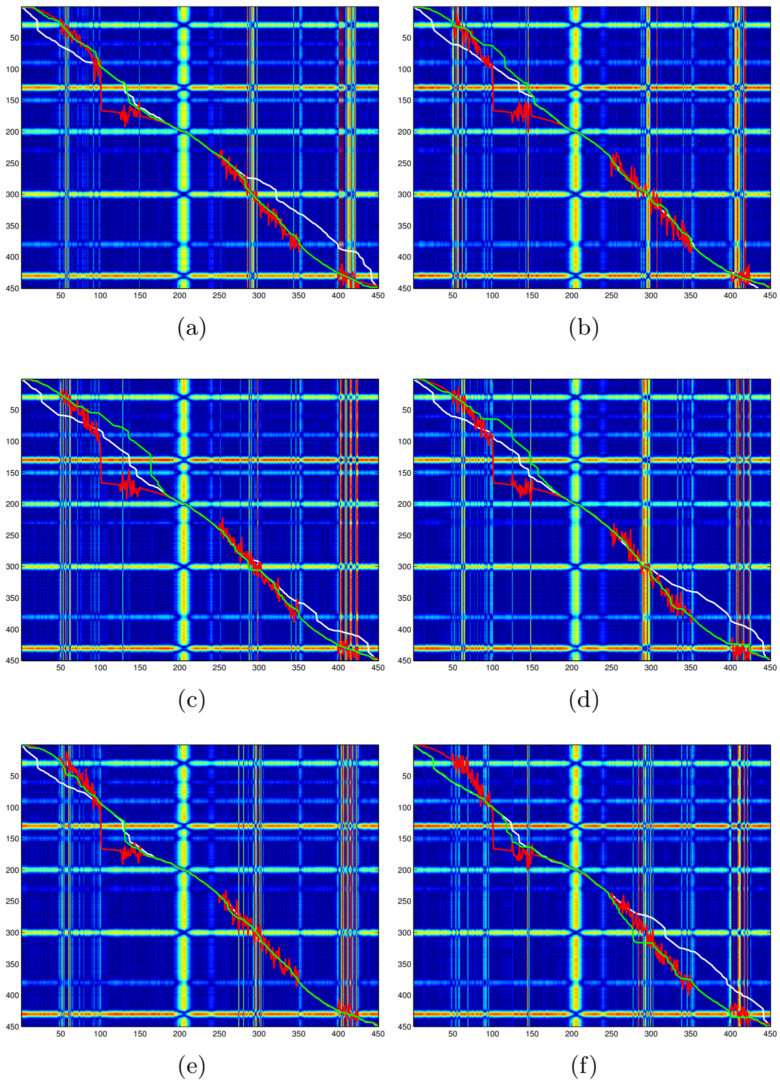
where B_i and E_i indicate the starting and ending time point of i^{th} non-causal interval. The non-causal time intervals are $[50,100], [125,150], [250,350]$ and $[400,425]$. For every time-series the contrasting sequence is generated by nearest neighbour interpolation at time points given by (5.60). A sample of a sequence and its non-causal warped version are shown is Figure 5.5. SPHMM parameters are learned using Alg. 1 for aligning every sequence and its warped (causal or non-causal) version. We tried segment lengths $l_x = l_y = [50,100,150,200]$. For a fair comparison with DTW we tried 10 different gap penalties (constant) from 0 to 100 which was applied for every gap operation. zero gap penalty yielded best result for DTW. Six of such alignments are depicted in Figure 5.6. The background is the distance between each sample. The ground truth given by (5.60) is plotted in red while the resulting alignment from DTW is is drawn in white and that of SPHMM in green. Both axes indicate time and plots are overlaid on the pairwise distance of the two sequences. It is obvious from Figure 5.6 that SPHMM outperforms DTW in aligning the non-causal time-series. To give a *quantitative* assessment of the goodness of the alignment, the ground truth is compared with reported correspondences by each algorithm. It should be noted that while DTW gives a correspondence for every time-point of the sequence, SPHMM produces segments. These segments are indicated by the starting and ending points. To be able to compare the sequence of segments with ground truth we have used linear interpolation. The goodness measure is the L_1 distance of every correspondence from the ground truth. The average L_1 distance for DTW over 100 alignments is 8258.8. This value is different for SPHMM for various segment lengths. Namely, the average distance is 7625.5, 5487.1, 5458.5, 5356.0 for $l_x = l_y = [50, 100, 150, 200]$ respectively. It is interesting that the distance does not change much for $l_X, l_Y > 100$. The reason is that the largest non-causal interval is 100 time-points long. In many cases the correct segments are extracted except for the second time interval which is located on the valley of the warping function where decoding the correct alignment is difficult for both algorithms.

5.6.2 Synthetic Data II

We used the same synthetic data in Section 4.4.1. The dataset is consisted of sinusoidal and rectangular signals which are embedded into Gaussian noise such that the placement of the signal is also random. Two samples of this dataset are shown in Figure 5.7. For IsoCCA experimentation we generated 10 samples from each class and used 1-NN classifier in a leave-one-out setting. We have shown that IsoCCA can achieve 90% accuracy while DTW cannot do better than 60%. We however, need to train SPHMM parameters which is not feasible using a training set derived from 20 sequences. Therefore, we generate 20 more sequences for training the parameters. SPHMM can classify the 20 sequences in test set with 100% accuracy. To make sure that the small size of the dataset is not affecting the result we generated 100 sequences and used 5-fold cross-validation setting. we observed that SPHMM is still able to perfectly classify all sequences.

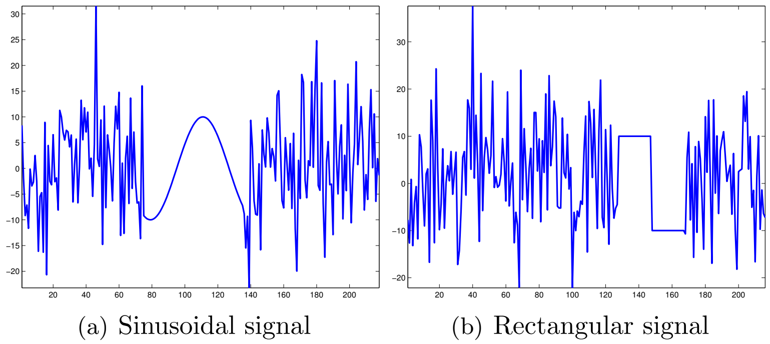
5.6.3 Benchmark Data

In order to compare our proposed approach to DTW and demonstrate the applicability of our method to general sequences, we tested SPHMM on the first subset of time-series from the UC Riverside time-series repository that contains 20 datasets. The length of time-series in this dataset varies from 60 to 637. To be able to test the noise resilience of SPHMM, we have added two types of noise to all sequences. The first noise model in well-known impulse noise. Impulse noise model is very well-known in signal processing community and can model abrupt sensor failure (or other rapid change effects) [65]. In particular, additive noise process is Gaussian $N(0, \omega\sigma_i)$ where σ_i is the standard deviation of feature i and ω is the power degree of the noise. We have added the noise to time points chosen uniformly at random such that the noise does not cover more than 20% of the sequence duration (Figure 5.8). We conducted the experiment



(a) (b) (c) (d) (e) (f)

Figure 5.6: Samples of aligning two sequences with non-causal intervals. Each plot depicts the comparison of the ground truth alignment (red) with DTW (white) and SPHMM (Green). The plots show the result for SPHMM with $l_x = l_y = 150$.



(a) Sinusoidal signal

(b) Rectangular signal

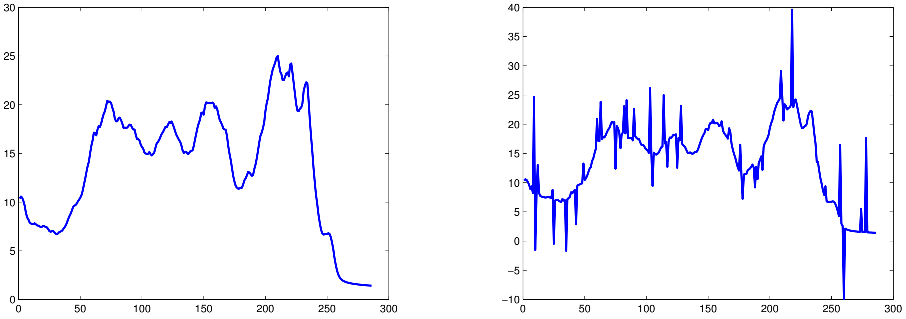
Figure 5.7: Synthetic data from 4.4.1.

on original data and noisy version of data with $\omega = 1$. For every sequence, we have generated three noisy samples (three noisy sequences) of the corresponding time-series. The algorithms (DTW, PHMM and SPHMM) are then applied to each noisy version of the data and the resultant recognition accuracies are averaged and reported. The results are shown in Tab 5.2, We compared the proposed approach to DTW and pair-HMM (where no segmentation is applied) with the warping band. To investigate whether DTW with a noise removal pre-processing is superior to SPHMM, we removed the noise using a median filter with fixed window size of 5 and showed the recognition rates in the DTW-NR column. We have applied the Skao-Chiba band suggested by UCR time-series page to DTW and PHMM. For SPHMM the maximum of the aforementioned band and twice the maximum segment length is chosen as the band to allow SPHMM accommodate up to two segments away from the diagonal of the alignment matrix. The parameters of SPHMM are learned using the method defined in Alg. 1. The segment length distribution however, is not learned and assumed to be uniform. In our experiments we noticed that the model is sensitive to segment length distribution and introducing a non-uniform prior can quickly lead to overfitting. This is due to the fact that the longer segments behave more like outliers. Therefore, it makes sense to use uniform as the segment length distribution. The parameters are not changed for noisy data experiments.

One can see in Table 5.2 that PHMM is superior to DTW in 9 cases and SPHMM is superior or on par with PHMM in all cases and superior to DTW in 15 cases in the original, noise-free setting. However, as soon as the noise is introduced, SPHMM shows significantly better performance compared to both DTW and PHMM even though PHMM still outperforms DTW. One may also notice that even though the median filter noise removal has elevated the recognition rates of DTW (DTW-NR column of impulse noise section in Table 5.2), it still falls behind SPHMM except for three cases. The superior performance of DTW-NR in those three cases is due to the fact that the window size of median filter accidentally matches the noise spread in one or two noisy versions of those datasets. However, there is no clear way of guessing the correct window size in advance.

To investigate whether the reported results indeed indicate the significance of SPHMM, we have performed Wilcoxon signed rank test [66]. In our case for a two-tailed Wilcoxon signed rank test on 20 datasets and $\alpha = .05$, $T = min(R^+, R^-) < 52$ was used to assert the significance of the proposed classifier[1]. Table 5.1 summarizes the results of significance testing. As one can observe SPHMM performs significantly better than other methods in all cases. In the original,

[1] R^+ (R^-) denote the total rank of datasets where the accuracy of method A is higher (lower) that the accuracy of method B. See [66] for details.



(a) Original

(b) $\omega = 100\%$

Figure 5.8: Sample of a sequence from UCR dataset (Coffee) with and without noise.

noise-free setting, PHMM's performance is superior to that of DTW but trails the performance of SPHMM. Since the significance of DTW-NR over DTW in the case of noisy data is very much evident, we have not reported this 5.1.

	Original		Impulse Noise	
	PHMM > DTW	**SPHMM > PHMM**	**DTW-NR > PHMM**	**SPHMM > DTW-NR**
R^+	141.5	141.5	181	141.5
R^-	31.5	5.5	10	36

Table 5.1: Wilcoxon signed rank test for Table 5.2. ">" stands for "significantly better". Boldface indicates statistically significant relationships.

The average length of the extracted matching segments is approximately 1.04 with a standard deviation of 0.20 in case of noise free data. For the noisy version of the dataset the average length of the matching segments rises to 1.60 with standard deviation of 1.01 indicating that many segments are detected. One has to note that since the chosen data does not result from the random delay processes, detecting many segments of lengths 1, i.e a sample-to-sample matching, is not unexpected.On the other hand, and due to noise (inherent or artificial), it is advantageous to have intermittently extended segments as evident from the reported standard deviation.

To demonstrate that our approach is resilient to well-known additive Gaussian noise, we have done the same experiment with the noise spread over the whole span of the signal. Since the noise is more dominant in this case the maximum segment length is increased to 10. We have performed noise-removal using and average filter before applying DTW to make sure that a noise removal with constant window size cannot improve the performance of DTW beyond SPHMM. The average filter window size is 10. The learned parameters are not changed from original case. The result is again reported in Table 5.2. The significance of SPHMM, is obvious and proved by Wilcoxon signed rank test depicted in Table 5.3. It is noteworthy that in the case of noisy data, pair-HMM is not significantly better than DTW when $\alpha = .05$, underlining the importance of longer segments extracted and matched by SPHMM. It is interesting to note that noise removal was not able to improve the the performance of DTW and furthermore, in 15 cases has caused a degradation of the performance. This is due to the constant window size and the fact that it does not adapt to the data which is crucial in case of such excessive noise. To assert this conclusion we picked "Trace" and "Adiac" dataset and tried different window sizes for filtering. The result showed significant improvement when the window size is set to 18 for "Trace" and 5 for Adiac. In particular, their accuracy improved to 82.31 and 12.12 for "Trace" and "Adiac", respectively. Another surprising point is that the accuracy results for Beef dataset is higher in noisy case putting the quality of this dataset in doubt (normalization removes this odd behaviour).

Computation Time: Figure 5.10 depicts the comparison of the average per alignment computation time between DTW and SPHMM when applied to original noiseless data. For short time-series the overhead of computing summed area table is dominant. For longer time-series the computation time is roughly 4 times that of DTW which is much better than the worst case. This is due to the fact that when the algorithm is investigating all segmentations for a correspondence for the first time, it has to find the score of a full alignment for every particular segment. This results in storing the score for every correspondence within all segments originated from that correspondence. Therefore, it is not necessary to recompute those values later when investigating the segmentations for neighbouring correspondences (neighbourhood is defined by the maximum segment length).

Spectral Analysis: Discrete Cosine Transform (DCT) [67] is a well-known tool for analyzing time signals [68, 69, 21]. In summary, DCT is an orthogonal linear transformation that expresses the signal with weighted summation of cosine functions with different frequencies. One can approximate the signal by selecting the cosines that constitute the major portion of the signal's energy, based on their computed coefficients, and thus compress it efficiently by storing only their properties such as frequency and coefficient. We define the complexity of a signal as the number of DCT components that are needed to reconstruct it properly. That is, the complexity of a signal increases with the number of DCT components that are required to represent it.

To understand where SPHMM is working better than DTW and where it does not, we looked at the DCT analysis of all signals in UCR time-series repository with no added noise. We averaged all time-series in each dataset separately regardless of their class labels and applied DCT with appropriate length on them. The number of DCT components that comprise 99% of

	Original			Gaussian Noise				Impulse Noise			
	DTW	PHMM	SPHMM	DTW	DTW-NR	PHMM	SPHMM	DTW	DTW-NR	PHMM	SPHMM
Lighting7	71.23	75.34	**79.45**	57.99	53.42	63.01	**68.03**	43.51	59.10	55.01	**73.94**
OSULeaf	61.98	65.7	**66.12**	41.32	53.99	63.77	**65.01**	47.11	55.55	55.15	**67.18**
OliveOil	83.33	**86.67**	**86.67**	**37.78**	27.78	35.56	35.56	28.89	**51.12**	28.89	32.22
SwedishLeaf	84.8	80.64	**85.28**	29.12	27.52	41.28	**53.01**	27.84	52.02	46.43	**57.81**
Trace	98	**100**	**100**	76.67	66.33	80.67	**81.67**	73.67	**89.93**	75.83	88.67
Two Patterns	99.33	**100**	**100**	94.66	96.24	95.36	**96.85**	88.22	99.85	89.86	**99.96**
fish	82.86	**86.86**	**86.86**	35.43	33.91	36.38	**38.47**	34.18	60.13	60.70	**71.21**
synthetic control	**98.67**	96.67	97.33	82.33	60.78	83.55	**85.33**	92.78	**98.33**	92.89	93.2
wafer	99.56	**99.76**	**99.79**	99.44	95.38	99.03	**99.79**	84.01	97.21	89.60	**99.39**
yoga	84.17	84.2	**84.23**	**78.19**	76.86	72.06	72.87	63.00	68.18	65.77	**77.18**
50words	77.14	80	**80.44**	29.08	70.18	70.48	**71.14**	57.21	74.12	74.12	**77.87**
Adiac	60.61	**60.87**	**60.87**	10.66	7.33	10.91	**14.41**	10.20	28.17	14.59	**40.04**
Beef	**53.33**	**53.33**	**53.33**	53.33	54.44	**55.55**	**55.55**	40.00	50.00	50.00	**53.33**
CBF	99.67	**99.89**	**99.89**	85.78	64.71	88.11	**88.74**	74.35	97.33	85.93	**98.01**
Coffee	82.14	78.57	**87**	65.47	70.24	60.71	**87**	57.14	73.81	63.22	**76.78**
ECG200	88	**91**	**91**	85	72.67	84.33	**86**	77.00	78.00	81.00	**85.00**
FaceAll	**81.72**	77.51	79.59	63.89	27.97	66.31	**72.25**	67.89	66.84	69.05	**77.20**
FaceFour	89.77	89.77	**92.05**	84.47	73.48	87.5	**90.15**	52.65	80.04	68.88	**89.07**
Gun Point	92	**98**	**98**	**76.22**	70.67	66.22	68.45	71.33	83.31	75.80	**84.65**
Lighting2	**86.89**	**86.89**	85.25	75.96	71.04	81.42	**83.61**	61.97	**87.43**	76.89	86.89
Average	83.76	84.58	85.66	63.135	58.74	67.11	70.69	57.1	73.20	65.98	76.43

Table 5.2: UCR time-series classification accuracy in presence of additive Gaussian and impulse noise models.

	DTW > DTW-NR	**SPHMM > DTW**	**SPHMM > PHMM**	DTW ≈ PHMM
R^+	160	167	190	142
R^-	50	27	1	55

Table 5.3: Wilcoxon signed rank test for Table 5.2 additive Gaussian noise section. ">" stands for "significantly better". Boldface indicates statistically significant relationships.

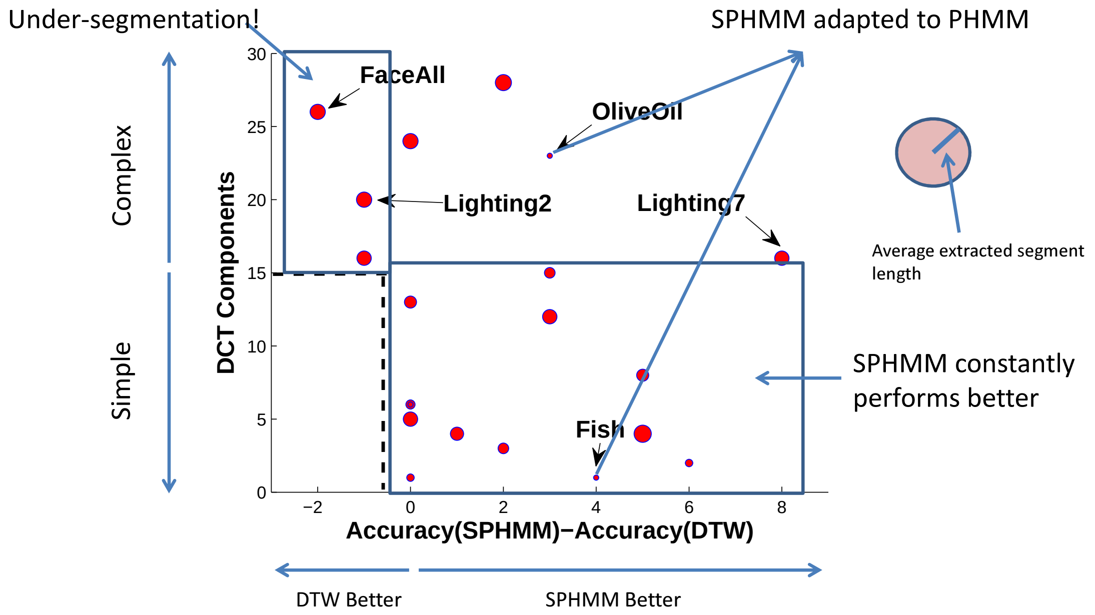


Figure 5.9: UCR time-series database DCT analysis. Horizontal axis shows the difference between the accuracy of SPHMM and DTW such that higher positive number indicates higher significance of SPHMM. Vertical axes shows the number of DCT components needed to reconstruct the average time-series. The radius of each disk is proportional to the average extracted segment length over all pairwise alignments between train and test set for that time-series.

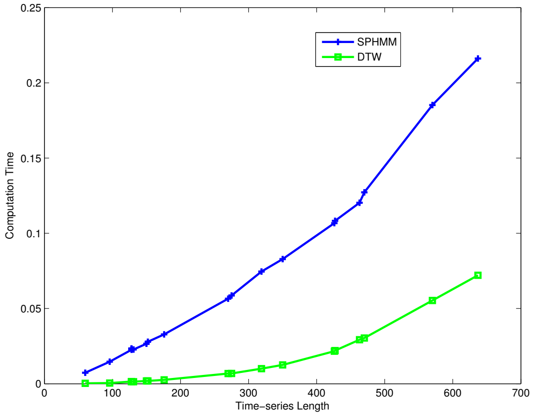


Figure 5.10: Comparison of the average per alignment computation time of SPHMM and DTW. Vertical axis show the time in seconds

the signal energy is retained. Figure 5.9 illustrates the relation between the complexity of the time-series, recognition rate of SPHMM compared to DTW and the average extracted segment length.

It is known in the literature that DTW and Euclidean distance are not very effective when the signal is very simple [70]. Looking at Figure 5.9, one can see that this is exactly where SPHMM constantly works better or at least in par with DTW. The only three datasets for which SPHMM works slightly worse than DTW are relatively complex signals. Note that in all those three cases the average segment length is relatively high. This shows that SPHMM tends to extract unnecessary long segments for complex signals. In fact, SPHMM might capture a component, which is naturally represented with a small variation in time domain, in a segment and match it to the wrong but very similar segment in the contrasting sequence. Also note that under-segmentation by SPHMM for complex signals is not always the case and it depends on other properties of the data such as variation among classes. For instance, *Olive Oil* is classified with much superior rate compared to DTW while being complex and also with smallest possible segment length, 1. To summarize, if the signal is not very noisy but is relatively complex, one might want to try smaller maximum segment lengths as it might result into a better performance.

5.6.4 EEG Signal Classification

We next applied our adaptive segmental alignment model to EEG signals to show its effectiveness in case of non-causal and noisy time-series. We used the P300 dataset described in [71]. Each subject is exposed to 6 different images, one of which is the target image. Dataset consists of 9 subjects. Four session are held for each subject. In each session six runs are conducted such

	$l = 1$		$l = 5$		$l = 10$		$l = 20$	
	Acc	St.dev	Acc	St.dev	Acc	St.dev	Acc	St.dev
SPHMM	74.7	2.61	75.1	2.97	78.47	2.35	82.64	1.35
DTW	74.4	1.78	N/A					
CTW	75.52	1.01	N/A					

Table 5.4: Recognition rates for EEG dataset. The first row shows the maximum segment length. For each maximum segment length the mean accuracy and standard deviation over different folds are reported.

$l_{fixed} = 5$		$l_{fixed} = 10$		$l_{fixed} = 20$		$l_{fixed} = 30$	
Acc	St.dev	Acc	St.dev	Acc	St.dev	Acc	St.dev
70.62	2.14	72.79	2.16	73.89	2.62	72.64	2.17

Table 5.5: Accuracy results for different fixed segmentations

that the set of all 6 images is shown at least 20 times to each subject where one of the images is the target in each run. We chose subject 1 and target 2 for our experiment. In each fold of cross-validation we keep one session as training and the remaining three are used as the test set such that every session is used as training once. 1-NN is used as the classifier. We applied the default pre-processing on the data except that we increased the sub-sampling rate to 128 from 32 to acquire longer signals (129 samples). As recommended, we only kept 8 channels. We have compared SPHMM against DTW and CTW [4]. The spatial embedding included by CTW is a reasonable choice for aligning EEG signals. We have applied SPHMM with different maximum lengths to demonstrate that the longer segments and permutation invariance of the distance metric can result in improved recognition rates.

The results are shown in Table 5.4. As expected the accuracy does not show significant improvement over DTW for maximum segment lengths of 5. However, for longer segments SPHMM becomes significantly more accurate. Optimal performance of DTW was achieved without a warping band.

We also applied our proposed forward algorithm approximation (5.16) to examine its performance and compare it to the dynamic programming. We tried segment length of 10 and the forward algorithm yielded 79.1(±1.12) which shows a marginal advantage for the marginal matching algorithm.

To assess the effects of adaptive segmentation and alignment we also tested against sequences pre-segmented into fixed length segments. The results are shown in Table 5.5. Adaptive segmentation remains advantageous especially for longer segment lengths.

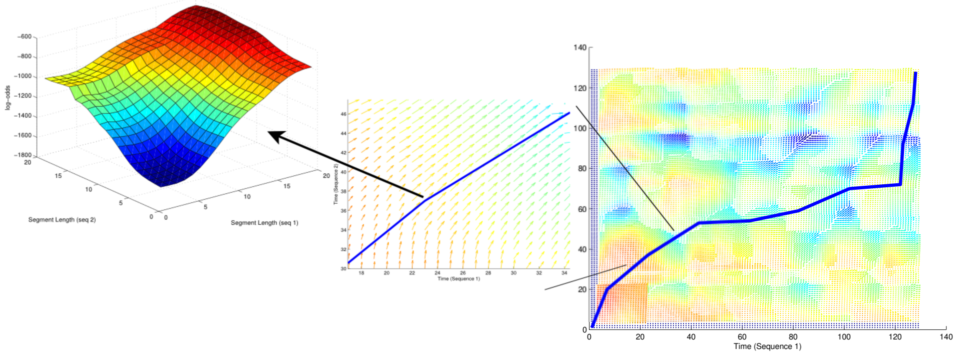


Figure 5.11: Segment length distribution for all positions for two EEG sequences. A certain portion of the graph is magnified. Smaller graph shows the likelihood of all possible segmentations for a single position (24,38) in the alignment matrix

Segment Length Distribution: Based on (5.17) we estimated the likelihood of all possible segmentations in aligning two EEG sequences for a maximum segment length of 30 and visualized it in Fig 5.11. The right-most graph depicts a vector field where each vector points to the most likely segment length (result of 5.17) at the corresponding position in the warping matrix and darker color indicates higher likelihood. The optimal alignment path is shown in the graph. A small portion of the graph is magnified in the middle graph, and then with the left-most graph depicting an example of the likelihood of all possible segmentations for a single position (24,38) selected by the alignment algorithm as a match operation. The chosen segment length at that position is 16 and 20 which has the highest likelihood and is the same segmentation selected by the alignment algorithm. This indicates the approximated forward algorithm can potentially be used to learn an improved local segmentation model.

Figure 5.12 shows the histogram of selected segment lengths for all pairs of sequences by aligning all recordings of two full sessions for target 2. The maximum segment length is set to length of the sequence to observe which segment lengths are selected without being limited to an upper bound. Since likely segments were mostly below the length of 20 we only show that potion of the histogram. Segment length of 1 and 1 is the most likely segment length. If this was not the case it would be very unlikely that DTW could result in any successful alignment.

5.6.5 Motion Capture Data

In order to show the effectiveness of our model in a challenging real-world application we performed experiments on two motion capture datasets. The first one is CMU-MoCap [54],

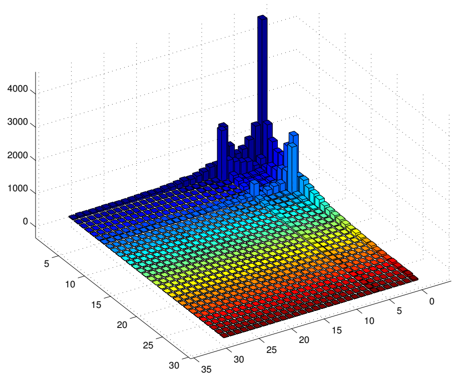


Figure 5.12: The distribution of segment lengths selected by alignment algorithm for all pairwise matches with maximum length of 129. Note that no segments of length over 30 were ever chosen.

which enables us to compare our results in this Chapter to IsoCCA. The second one is HDM05 MoCap dataset [72].

CMU MoCap: To contrast our approach with IsoCCA we tested SPHMM on MoCap sequences in the same setting. We used the same selection of sequences as in Section 4.4.2. Namely, 62 sequences containing more than 40000 frames of 8 different actions from CMU MoCap dataset. walking, runing, boxing, jumping, marching, dancing, sitting down and shaking hands. Each class contains 7, 10, 8, 6, 10, 10, 7 and 4 sequences, respectively. Classes were selected with actions performed by different subjects. The dimensionality of data is reduced from 62 to 10 using PCA while keeping 99.8% of the energy. We compared SPHMM to IsoCCA, DTW and CTW [4]. 1-NN is used as the classifier to find the closest sequence to any given query in a leave-one-out setting. Parameters for SPHMM are empirically set to $\delta = 0.001, \epsilon = .1, \tau = 0.01$ and $l_1 = l_2 = 10$. In DTW Sakoe-Chiba constraint with $\rho = 13\%$ is imposed to improve its performance in classification. For higher levels of noise we have permitted more gap operations for DTW by increasing warping window to $\rho = 18\%$. CTW is applied on the original 62 dimensional data set as it showed a better performance on it. As mentioned in [6], CTW is unable to achieve better results than DTW. The recognition accuracies are shown in Table 5.7.

Our method shows significantly higher performance compared to the other methods. The segmental approach was able to recognize proper segments of sequences and match them to their corresponding segments on the contrasting sequence. As an example, in Figure 5.13, we have shown a portion of the alignment of two boxing sequences. Segments are separated by red lines and matched segments are indicated by arrows. Segments with no arrow pointing to them are either deleted or inserted based on the sequence one may take as reference. One can

Table 5.6: Accuracy of fixed segmentation

	$l_{fixed} = 5$	$l_{fixed} = 10$	$l_{fixed} = 20$	$l_{fixed} = 50$
Accuracy	80.65	74.19	77.42	69.35

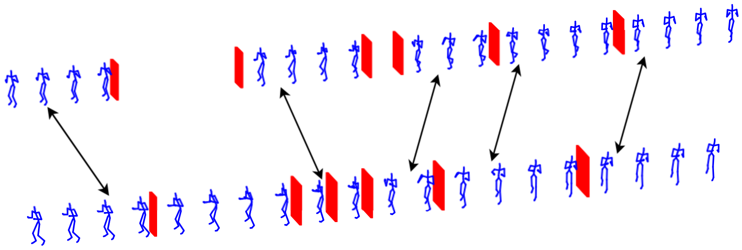


Figure 5.13: A portion of alignment of two boxing sequences. Segments are separated by red lines. The matched segments are indicated by arrows. Those segments with no arrow pointing to them are either deleted or inserted.

observe that similar actions are distinguished and matched. This can be explained by the fact that if the two partitions are similar and do not change drastically, the segment length tends to be longer (ref. Section 5.4). Another interesting observation is that the direction of action is ignored. Last match depicted in the figure, shows the correspondence of two punching actions one in forward and the other one in backward direction. In an action recognition task one is typically interested in retrieving actions regardless of their direction. However, the change of direction can sometimes introduce practical difficulties.

Average match segment length for MoCap was 3.70 with standard deviation of 4.05 showing that many (relatively) long segments are selected. Again to assert the efficacy of adaptive segment length determination we compared our main results against fixed segmentation (Table 5.6). The results are significantly inferior to adaptive SPHMM. Based on table 5.6 we assume that adaptive segmentation with maximum segment length of 20 may result in an even a better performance.

To assess the noise resilience of the SPHMM compared to other methods we added impulse noise in the same way described in Section 5.6.3 except that the spread of the noise is restricted to 5% of the sequence. The noise is added only to the query sequences and the experiment setting is as above. To investigate whether a noise removal pre-processing can improve the performance

Table 5.7: Accuracy of SPHMM versus other methods

	SPHMM	IsoCCA	DTW	CTW
Accuracy	90.32	87.10	82.26	50.64

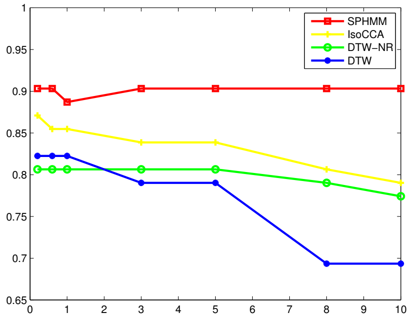


Figure 5.14: Comparing recognition accuracy of SPHMM versus other methods in presence of noise. Horizontal axes shows the level of noise.

of DTW beyond SPHMM, we apply a median filter on the data and show its performance with DTW-NR along with the accuracies of DTW, IsoCCA and SPHMM in Figure 5.14. The noise level in Figure 5.14 starts form $\omega = .2$ to make the noise removal performed on the query for DTW more meaningful. Obviously, noise removal on clean data will result in loss of information and leads to degraded performance for DTW. One can observe the stability of the classification accuracy of SPHMM in presence of different levels of noise. The noise removal can elevate the performance of DTW at high noise levels but it reduces the accuracy in lower levels of noise.

HDM05: Contains MoCap data which is consisted of 2-3 rotation angles of 29 skeletal joints, resulting in 62 joint angle time series. HDM05 includes 100 classes of action performed by 5 subjects. We choose 8 action classes which are *DepositFloorR, JumpingJack, KickRFront, KickRSide, PunchLFront, PunchRFront, Squat, Walk2Steps.* Sequences are around 300 time-points long and the whole dataset contains 276 sequences in total. We perform 5-fold cross validation and 1-NN is our classifier. Maximum segment length is set to 10. We compare our method against DTW, canonical time warping (CTW) [4] and IsoCCA. SPHMM achieved the highest accuracy, 85.5(±6.18). DTW, CTW and IsoCCA yield 70.1(±5.09), 60.2(±5.1) and 75.1(±6.8) respectively. The significance of SPHMM is evident from the reported results. The confusion matrix for this experiment is shown in Table 5.8. One can notice that *DepositFloorR* is confused with *Walk2Steps* and *KickRFront* with *KickRSide*. It should be noted that *DepositFloorR* contains the action of walking (one or two steps) right before actual depositing. Also *KickRFront* and *KickRSide* are very much alike. *PunchRFront* is also sometimes confused with *KickRFront*, *KickRSide* and *PunchLFront* where one can perceive that those actions have a lot in common making it difficult to distinguish them correctly in some instances.

	DFR	JJack	KRF	KRS	PLF	PRF	Sq	W2S
DepositFloorR	65.6	0	0	0	6.3	3.1	0	25
JumpingJack	0	98	0	0	0	0	0	2
KickRFront	0	0	75.9	20.1	0	0	0	3.5
KickRSide	0	0	21.40	71.4	0	3.6	0	3.6
PunchLFront	0	0	3.6	3.6	82.1	10.7	0	0
PunchRFront	0	6.7	0	6.7	6.7	80	0	0
Squat	0	0	0	0	0	0	100	0
Walk2Steps	0	3.5	3.5	0	0	0	0	93.1

Table 5.8: Confusion matrix of action recognition for SPHMM(in percentage points)

Figure 5.15: Sample frames from UT-interaction dataset #1.

5.6.6 UT-Interaction

To apply segmental matching we needed to pick a dataset of reasonable length and complexity so we could try different segmentation lengths and observe how the recognition rate is affected. Therefore, popular action recognition datasets such as KTH [73] or Weizmann [74] datasets were not suitable for our settings because they contain short periodic actions and only a few frames are sufficient for a reliable recognition. Instead, we use the first subset of publicly available UT-interaction dataset containing 10 sequences (60 after segmentation of actions). Within each sequence, six actions, *hand shaking, hugging, kicking, pointing, punching* and *pushing* are performed by 10 different actors. The videos involve camera jitter. Pedestrians are present in the video which makes the recognition more difficult (Figure 5.15). We have used spatio-temporal interest points (Cuboids) [75] as the descriptors. Then k-means is applied on the resulting features to produce an 800 element codebook.

We use a nearest neighbour classifier to compare with [5]. Leave-one-sequence-out cross-validation by holding one sequence for testing and using the remaining nine for training. Each action in the test set is matched with all training sequences. As a baseline we report the results on SVM using the same feature set and also the results reported in [5]. We have used l_1 and χ^2 histogram distances. The results on the l_1 distance metric are reported in Table 5.9. It is evident from the results that our approach significantly outperforms other methods. Using either l_1 or χ^2 distance metrics SM and Fast-SM were able to achieve the best result when the maximum segment length was 30. χ^2 achieved the best result even with maximum segment length of 20. We tried different maximum segment lengths, namely, 10,15,20, 25 and 30. Figure

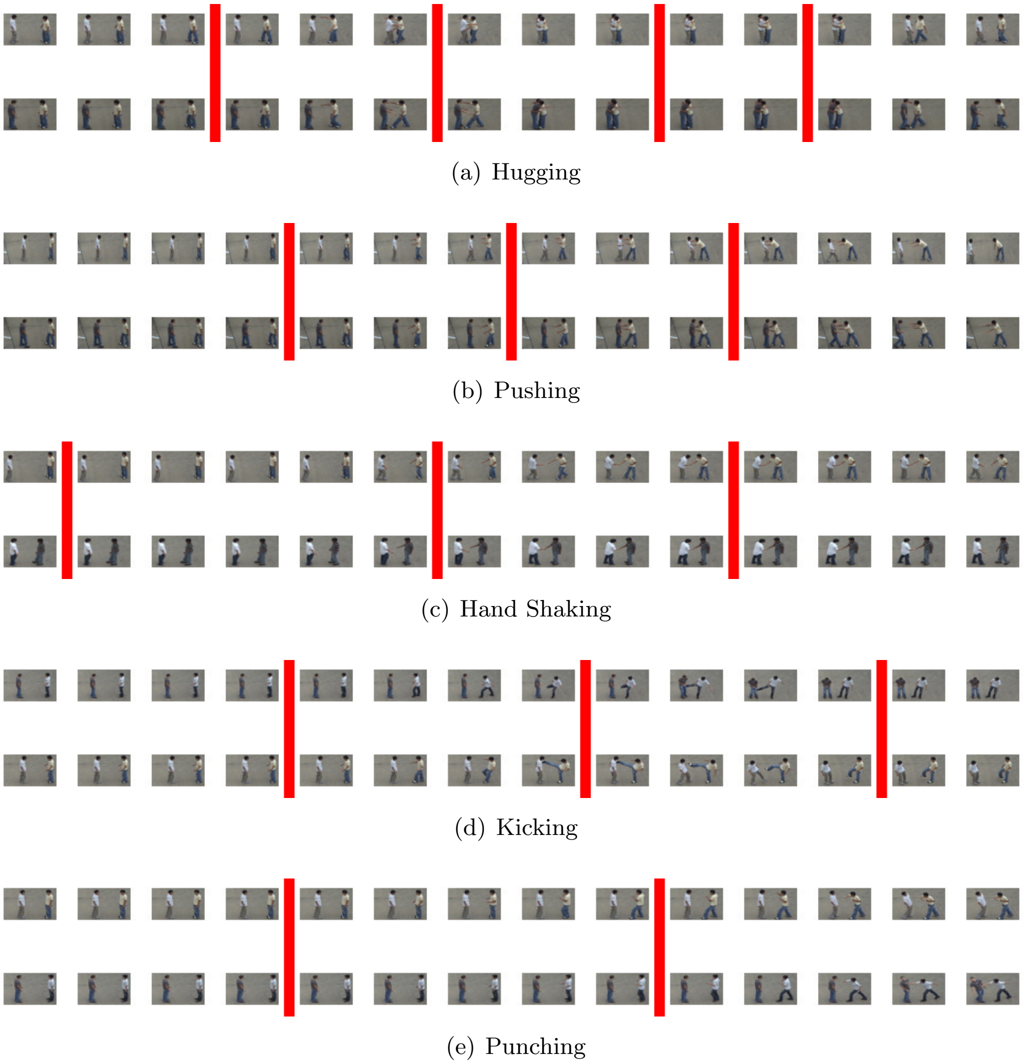
5.17 illustrates how the resulting accuracy and speedup, gained by bounding the distance (Fast-SM), change as the maximum segment length increases applying l_1 and χ^2 histogram distance metrics. It is interesting to note that the recognition rates of Fast-SM and SM are identical in all cases eliciting the fact that the bounding technique and the smoothness assumption on the local likelihoods are in fact effective. In addition, Fast-SM achieves at least a 2-fold speedup compared to SM. As shown in 5.17(a), χ^2 achieves better results in smaller maximum segment lengths pointing to it as a more suitable measure of distance on segment histograms. Unfortunately, as the maximum segment length increases the bounds on the histogram distances become looser, resulting in reduced speedup. However, one should notice that the shortest sequence is 24 frames long and our final maximum segment length (30) already exceeds this limit. This implies that the model has the option to effectively considers a single BOTW representation as an alternative.

We also applied SPHMM to observe whether a complete alignment model is able to achieve better performance compared to SM and Fast-SM. The result showed that SPHMM cannot advance the recognition rate beyond 91.57% yet is at least three times slower than SM and four times slower than Fast-SM.

Samples of the discovered segments are depicted in Figure 5.16. Five activities are illustrated and each segment is separated using a red bar. Only a few frames from each segment is shown. The number of frames shown in each segment is proportional to the length of the segment such that a longer segment is shown with more frames comparing to a shorter segment in the same segmental alignment. An important observation is that the algorithm tends to encapsulate similar relative motions within each segment. For instance, in the 'Hugging' activity (Figure 5.16(a)), the second and the third segments, which both had the maximum length, encompass the action of hugging. The next segment, shorter in length, contains the pause when the two actors do not move substantially, while the last segment collects the frames corresponding to the actors separating from each other. One can speculate that the second and third segments would merge if the maximum segment length was large enough. However, having larger maximum segment length results in longer running time.

A disadvantage of forcing a fixed segmentation is evident from this result. Fixed segmentation makes it quite probable for more (or less) than one part of an activity to fall into a segment in one sequence and thus result into a sub-optimal matching to the corresponding segment in the contrasting sequence. This will in turn, results into a sub-optimal similarity measurement.

The distributions of discovered segment lengths, when the distance metric is χ^2, are illustrated in Figure 5.18. Very similar results have been observed for the l_1 distance metric. It is evident from the results that many segments with maximum possible length are discovered.



(a) Hugging

(b) Pushing

(c) Hand Shaking

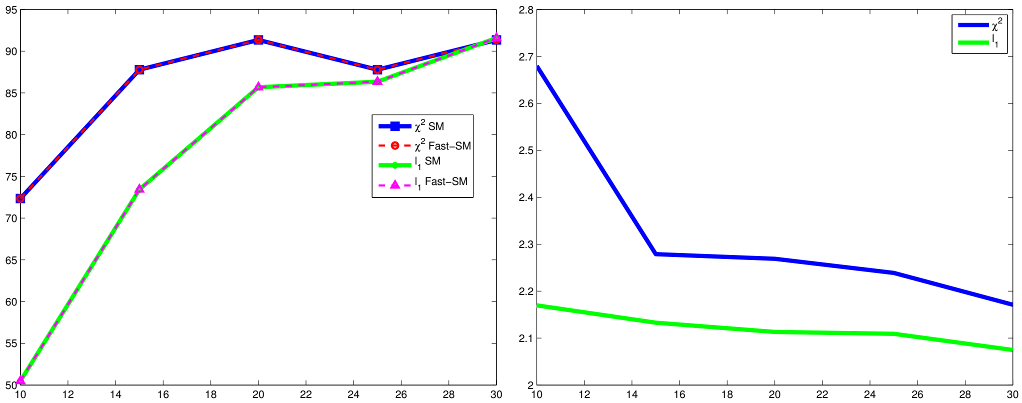
(d) Kicking

(e) Punching

Figure 5.16: Samples of discovered segments. Segments are separated by red bars. Only a few frames from each segment are shown. The segments and sequences are not necessarily of the same length. The number of frames shown for each segment is increased or decreased for better illustration.

Method	Accuracy
Segmental Match	**91.57%**
Dynamic BOW [5]	85.0%
SVM	85.0%
Voting [76]	88.0%

Table 5.9: Recognition rates on UT-interaction dataset #1.



(a) Accuracy

(b) Speedup

Figure 5.17: Accuracy and speedup results for l_1 and χ^2 distances. l_1 is depicted as green and χ^2 as blue. Accuracy result of Fast-SM for distance metric is identical to SM.

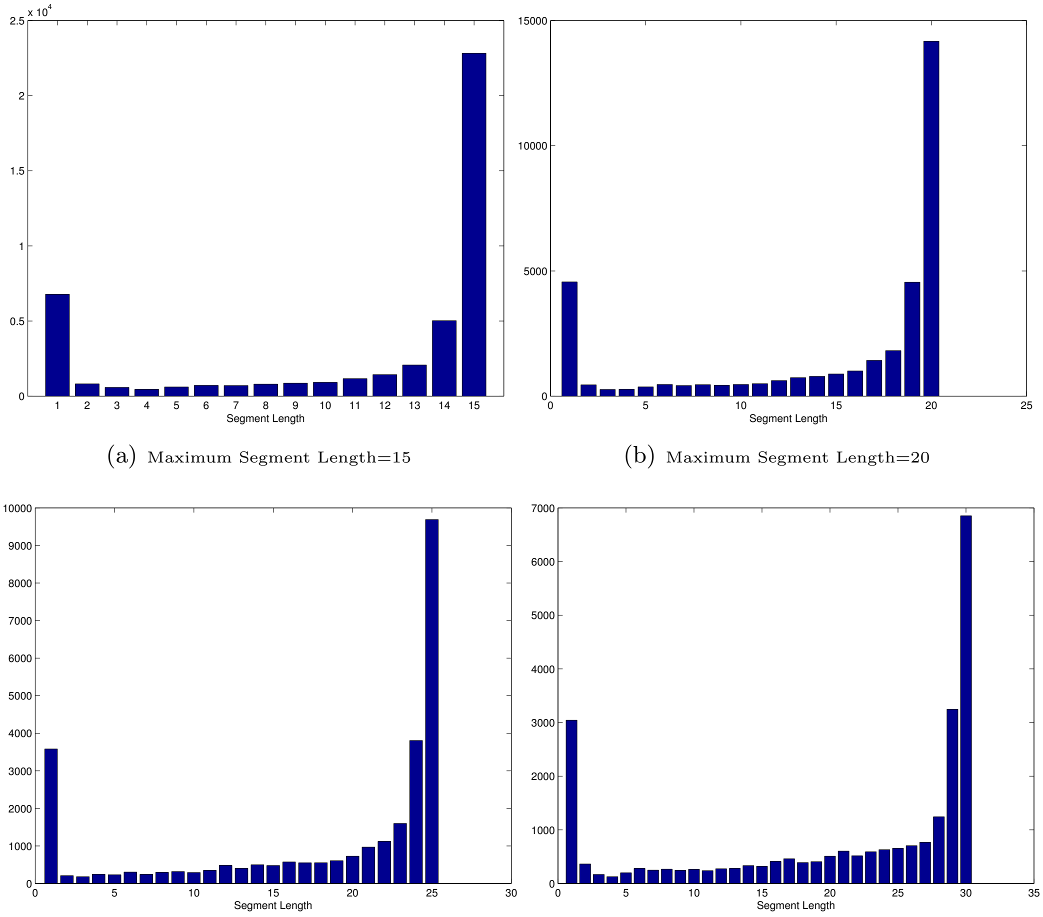
Accuray	Method
Fast-SM	71.67
SVM [77]	70.00
Hough Voting [76]	77.00

Table 5.10: Accuracy results on UTI Part #2

Also it is interesting to note that as the maximum segment length increases the shape of the distribution does not change, with one peak at each end of segment length range.

The proposed algorithm (Fast-SM) is applied on the second part of UT-interaction data where the activities are performed on a lawn in a windy day. The classification task is much more challenging compared to the first part since the background (trees and grass) is moving and also the camera jitter is more severe. The experimental setting is exactly the same as that of Part #1 reported in the main manuscript. The best result is reported in Table 5.10. The only results that is better than Fast-SM is reported in [76] where the authors use ground truth bounding boxes to eliminate many noisy features. The accuracy result of Dynamic BoW [5] on this dataset is not mentioned in the original paper. However, the rate reported in Figure 7 in [5] is less than 70%.

The resulting accuracy from using different distance metrics for four different maximum segment lengths are illustrated in Figure 5.19. Similar to the results reported on Part #1, one can observe consistently better performance of the algorithm when χ^2 is used as the distance metric over the l_1 distance measure.



(a) Maximum Segment Length=15

(b) Maximum Segment Length=20

(c) Maximum Segment Length=25

(d) Maximum Segment Length=30

Figure 5.18: Distribution of discovered segment lengths.

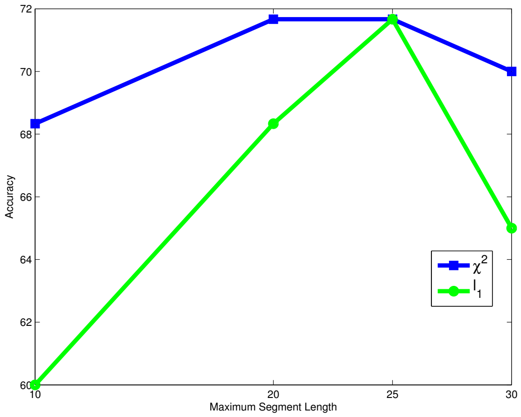


Figure 5.19: Accuracy of Fast-SM on second subset of UT-Interaction dataset. Blue line depicts results when χ^2 is the distance metric and green line is that of l_1.

5.7 Conclusion

In this Chapter we presented a probabilistic model for segmental sequences alignment. We showed that a modified pair-HMM, in conjunction with a proper segment metric, can lead to effective joint segmentation and segmental alignment. Our experimental results showed high accuracy particularly when confronted with high levels of noise where DTW does not perform well even after noise removal pre-processing. Additionally, the invariance to local permutation has enabled our algorithm to perform well on non-causal signals. We also proposed a relaxation over the original model which reduced the computational time. Especially when histograms were used to represent the time-series we were able to prune the unnecessary computation using bounds on histogram distance metrics.

Chapter 6

Conclusion

Time-series retrieval is a central problem to many recognition tasks. Even though a fast time-series retrieval technique is an essential part of many tasks, the correctness of such system is of crucial importance. High levels of noise and other artifacts that might be added to a time-series can make the problem even more difficult. Conventional similarity measures perform sub-optimally when the signal is corrupted by excessive amounts of noise and local shuffling of time samples. This results in an inferior performance of a time-series retrieval system.

In this work we introduced the concept of segmental alignment. We proposed to segment the sequences into short sub-sequences and measure the distance of contrasting time-series based on those segments. We proposed two methods for jointly realizing the segment boundaries and measuring the similarity of two time-series.

The first method (IsoCCA) was an isotonic regression version of canonical correlation analysis (CCA). We modified the original objective of CCA to impose the convexity constraint on the segments' coefficients and thus measure the distance of two segments as the closest distance of two convex-hulls. We further imposed another constraint to enforce the time monotonicity in the segment level.

The second proposed method (SPHMM) was an extension of a probabilistic alignment model, pair-HMM. We proposed a proper and efficient distance metric between segments. The observation model of the HMM was changed to accommodate for the segments and an efficient inference algorithm was proposed to jointly recover the segments and the likelihood of aligning two time-series. To increase the computational efficiency, we proposed a relaxation to the original model and combined it with a bounding method. The results showed superior performance of our method compared to the state of the art in a broad range of applications and publicly available datasets and benchmarks.

Choosing a similarity measure for time-series depends on the nature of the data and the requirements of the application. Tables 6.1 and 6.2 summarize our conclusion on the choice of similarity measure in a retrieval system based on the efficiency and accuracy. In Table 6.1, we

Similarity Measure	Misalignment	Noise	Local Non-causality
Euclidean Distance	No	No	No
DTW/pair-HMM	Severe	Low	No
LCSS/EDR/Swale	Severe	Moderate	No
SM/Fast-SM	Moderate	Moderate	Yes
IsoCCA/SPHMM	Severe	Severe	Yes

Table 6.1: Qualitative comparison of similarity measures based on the time-series data at hand.

Similarity Measure	Time Complexity and Running Time
Euclidean Distance	$O(N)$
Dynamic Programming	$O(NM)$
IsoCCA	$O(\frac{\max(N,M)^3}{\min(N,M)^2})$
SPHMM	$O(\max(l_x, l_y)NM)$
SM/Fast-SM	$O(\max(l_x, l_y)NM)$

Table 6.2: Comparison of similarity measures based on computational complexity. N and M are contrasting time-series' lengths. We have assumed the a pairwise distance is pre-computed for all methods. l_x and l_y are maximum segment lengths in SPHMM. SM and Fast-SM have the same worst case time complexity as SPHMM but their running time is at least 3 times better than that of SPHMM. The running time of IsoCCA is heavily dependent on the least square solver and the convergence rate of the optimization.

have chosen three properties of a dataset and report our observations on the ability of different similarity measures to deal with them. We have considered the severity of misalignment and noise and presence or possibility of local non-causality. IsoCCA and SPHMM are the most general algorithms and can cope with severe misalignment, noise and local non-causality. We however, pay a price in terms of required computation for more sophisticated methods. The computational complexity of alignment algorithms are summarized in Table 6.2. Segmental Matching and Fast Segmental Matching have the same complexity as that of SPHMM but due to pruning and relaxed model their actual running time is at least 3 times better than SPHMM.

Other considerations might also influence the choice of similarity measures when the dataset contains millions of time-series. In that case one has to make sure that fast and tight lower bounding methods exist for the similarity measure so that most of the comparisons need not to be even computed.

References

[1] G. W. Schwert, "Why does stock market volatility change over time?" *Journal of Finance*, vol. 44, no. 5, pp. 1115–1153, 1989. 1

[2] B. LeBaron, W. B. Arthur, and R. Palmer, "Time series properties of an artificial stock market," *Journal of Economic Dynamics and Control*, vol. 23, no. 9, pp. 1487–1516, 1999. 1

[3] D. Wang, B. Podobnik, D. Horvatić, and H. E. Stanley, "Quantifying and modeling long-range cross correlations in multiple time series with applications to world stock indices," *Physical Review*, vol. 83, no. 4, p. 046121, 2011. 1

[4] F. Zhou and F. de la Torre, "Canonical time warping for alignment of human behavior," *Advances in Neural Information Processing Systems (NIPS)*, pp. 1–9, 2009. 1, 8, 14, 27, 28, 34, 35, 36, 68, 70, 72

[5] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," *Proceeding of IEEE Conference on Compupter Vision*, pp. 1036–1043, 2011. 1, 41, 73, 75, 76

[6] S. Shariat and V. Pavlovic, "Isotonic CCA for Sequence Alignment and Activity Recognition," *Proceeding of IEEE Conference on Compupter Vision*, pp. 2572–2578, 2011. 1, 5, 70

[7] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Conf. on Knowledge Discovery and Data Mining (KDD)*, 2000, pp. 285–289. 1

[8] D. Goldin and P. C. Kanellakis, "On similarity queries for time-series data: Constraint specification and implementation," in *Conf. on Principles and Practice of Constraint Programming*, 1995, pp. 137–153. 1

[9] J. de Munck, S. Gonalves, L. Huijboom, J. Kuijer, P. Pouwels, R. Heethaar, and F. L. da Silva, "The hemodynamic response of the alpha rhythm: An EEG/fMRI study," *NeuroImage*, vol. 35, no. 3, pp. 1142 – 1151, 2007. 4, 26

[10] J. Prescott and J. Hutton, "Cosmic ray contributions to dose rates for luminescence and esr dating: large depths and long-term time variations," *Radiation Measurements*, vol. 23, no. 2, pp. 497–500, 1994. 4

[11] S. Shariat, V. Pavlovic, T. Papathomas, A. Braun, and P. Sinha, "Sparse dictionary methods for eeg signal classification in face perception," in *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, 2010, pp. 331–336. 4

[12] S. Shariat and V. Pavlovic, “Improved sequence classification using adaptive segmental sequence alignment,” *Journal of Machine Learning Research - Proceedings Track*, vol. 25, pp. 379–394, 2012. 5

[13] ——, “Robust time-series retrieval using probabilistic adaptive segmental alignment,” *IEEE Trans. on Knowledge and Data Engineering*, 2012. 6

[14] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis. Probabilistic model of proteins and nuclear acids.* Cambridge University Press, 1997. 7, 14, 43, 47

[15] J. Listgarten, R. M. Neal, S. T. Roweis, and A. Emili, “Multiple alignment of continuous time series,” in *Advances in Neural Information Processing Systems (NIPS).* MIT Press, 2005, pp. 817–824. 8

[16] M. Kim and V. Pavlovic, “Discriminative learning of mixture of bayesian network classifiers for sequence classification,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, pp. 268–275. 8

[17] S. Akimoto and N. Suematsu, “A nonparametric bayesian approach to time series alignment,” in *Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on*, 2010, pp. 648–653. 8

[18] W. C. K. I. Rasmussen C. E., *Gaussian Processes for Machine Learning.* MIT Press, 2006. 8

[19] Y. Chen, M. A. Nascimento, B. C. Ooi, and A. K. Tung, “Spade: On shape-based pattern detection in streaming time series,” in *IEEE 23rd International Conference on Data Egineering (ICDE).* IEEE, 2007, pp. 786–795. 8, 16

[20] D. Gamerman and H. F. Lopes, *Markov Chain Monte Carlo.* Chapman and Hall, 2006. 8

[21] 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*, vol. 1, no. 2, pp. 1542–1552, 2008. 9, 16, 27, 58, 64

[22] F. Zhou and F. De la Torre, “Generalized time warping for multi-modal alignment of human motion,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 10, 14

[23] D. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” in *KDD Workshop*, 1994, pp. 359–370. 10, 13, 26

[24] H. Sakoe and C. Chiba, “Dynamic Programming Algorithm Optimization for Spoken Word Recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978. 10, 12, 27

[25] M. J. Atallah and S. Fox, *Algorithms and Theory of Computation Handbook*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1998. 10

[26] L. Rabiner, A. Rosenberg, and S. Levinson, “Considerations in dynamic time warping algorithms for discrete word recognition,” *IEEE Trans. Acoustics, Speech, and Signal Proc.*, vol. ASSP-26, pp. 575–582, 1978. 13

[27] C. Myers, L. Rabiner, and A. Rosenberg, “Performance tradeoffs in dynamic time warping algorithms for isolated word recognition,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 28, no. 6, pp. 623–635, 1980. 13

[28] R. L. and J. B., *Fundamentals of speech recognition.* NJ, Printice Hall: Englewood Cliffs, 1993. 13

[29] E. J. Keogh and M. J. Pazzani, “Derivative dynamic time warping,” in *In First SIAM International Conference on Data Mining (SDM2001*, 2001. 13

[30] D. R. Hardoon, S. Szedmak, and J. Shawe-taylor, “Canonical correlation analysis ; An over view with application to learning methods,” *Neural Comput.*, vol. 16, no. 12, pp. 2639—-2664, 2004. 14, 29, 32

[31] B. Fischer, V. Roth, and J. M. Buhmann, “Time-series alignment by non-negative multiple generalized canonical correlation analysis.” *BMC bioinformatics*, vol. 8, p. S4, Jan. 2007. 14, 28

[32] S. Batzoglou, “The many faces of sequence alignment,” *Briefings in Bioinformatics*, vol. 6, no. 1, pp. 6–22, 2005. 14, 15

[33] H. Andre-Jonsson and D. Z. Badal, “Using signature files for querying time-series data,” in *First European Symposium on Principles of Data Mining and Knowledge Discovery*, 1997, pp. 211–220. 15

[34] M. Vlachos, G. Kollios, and D. Gunopulos, “Discovering similar multidimensional trajectories,” in *Data Engineering, 2002. Proceedings. 18th International Conference on*, 2002, pp. 673–684. 15

[35] L. Chen and R. Ng, “On the marriage of lp-norms and edit distance,” in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, ser. VLDB ’04. VLDB Endowment, 2004, pp. 792–803. 15

[36] L. Chen and M. T. zsu, “Robust and fast similarity search for moving object trajectories,” in *In SIGMOD*, 2005, pp. 491–502. 15

[37] M. Vingron and M. S. Waterman, “Sequence alignment and penalty choice,” *Journal of Molecular Biology*, vol. 235, no. 1, pp. 1–12, Jan. 1994. 15

[38] M. D. Morse and J. M. Patel, “An efficient and accurate method for evaluating time series similarity,” in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, ser. SIGMOD ’07. New York, NY, USA: ACM, 2007, pp. 569–580. 16

[39] J. Aßfalg, H.-P. Kriegel, P. Kröger, P. Kunath, A. Pryakhin, and M. Renz, “Similarity search on time series based on threshold queries,” in *Advances in Database Technology-EDBT*. Springer, 2006, pp. 276–294. 16

[40] M. C. Moulson, B. Balas, C. Nelson, P. Sinha, and C. Sceinces, “EEG Correlates of Categorical and Graded Face Perception,” *Journal of Vision*, vol. 8, no. 6, p. 533, 2008. 17

[41] S. Makeig, A. Bell, T. Jung, and T. e. a. Sejnowski, “Independent component analysis of electroencephalographic data,” *Advances in Neural Information Processing Systems*, p. 145151, 1996. 17

[42] A. Yazdani, T. Ebrahimi, and U. Hoffmann, “Classification of EEG Signals Using Dempster Shafer Theory and a K-Nearest Neighbor Classifier,” *Signal Processing*, pp. 327–330, 2009. 18

[43] S. Makeig, “Auditory event-related dynamics of the EEG spectrum and effects of exposure to tones.” *Electroencephalography and Clinical Neurophysiology*, vol. 86, no. 4, pp. 283–93, April 1993. 18

[44] R. Tomioka, S. Lemm, and M. Kawanabe, “Optimizing Spatial Filters for Robust EEG Single-Trial Analysis,” *IEEE Signal Processing Magazine*, pp. 41–56, 2008. 18

[45] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. 19

[46] J. Friedman, T. Hastie, and R. Tibshirani, “Regularization paths for generalized linear models via coordinate descent,” *Department of Statistics, Stanford University, Tech. Rep*, pp. 1–22, 2008. 19, 20, 22

[47] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal-Royal Statistical Society Series B Statistical Methodology*, vol. 68, no. 1, p. 49, 2006. 20

[48] V. Roth and B. Fischer, “The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms,” in *Proceedings of the 25th International Conference on Machine learning*. ACM, 2008, pp. 848–855. 22

[49] M. Kim and V. Pavlovic, “A recursive method for discriminative mixture learning,” in *Int’l Conf. Machine Learning (ICML)*, 2007. 27

[50] C. Ratanamahatana and E. Keogh, “Making time-series classification more accurate using learned constraints,” in *Proceedings of SIAM International Conference on Data Mining*. Lake Buena Vista, Florida, 2004, pp. 11–22. 27

[51] A. Veeraraghavan and A. K. Roy-Chowdhury, “The Function Space of an Activity,” *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR’06)*, pp. 959–968, 2006. 27

[52] A. Smola, A. Gretton, L. Song, and B. Scholkopf, “A Hilbert space embedding for distributions,” in *Algorithmic Learning Theory*, 2007, pp. 13–31. 28

[53] C. Lawason and R. Hanson, *Solving Least Squares Problems*. Prentice-Hall, 1974. 33, 34

[54] http://mocap.cs.cmu.edu/. 36, 69

[55] P. H. C. Eilers, "Parametric time warping." *Analytical chemistry*, vol. 76, no. 2, pp. 404–11, Jan. 2004. 37

[56] A. Woznica, A. Kalousis, and M. Hilario, "Distances and (indefinite) kernels for sets of objects," in *International Conference on Data Mining*, dec. 2006, pp. 1151 –1156. 42

[57] R. Kondor, "A kernel between sets of vectors," in *Proceedings of International Conference on Machine Learning*, dec. 2003. 42

[58] F. C. Crow, "Summed area tables for texture mapping," *Proceedings of the 11th annual conference on Computer Graphics and Interactive Techniques*, pp. 207–211, 1984. 47

[59] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE 77 (2)*, 1989, pp. 257–286. 49

[60] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2129–2142, Dec. 2009. 53

[61] H. Riemenschneider, M. Donoser, and H. Bischof, "Bag of optical flow volumes for image sequence recognition," in *British Machine Vision Conference*, 2009. 53

[62] W.-S. Chu, F. Zhou, and F. D. la Torre, "Unsupervised temporal commonality discovery," *European Conference on Computer Vision (ECCV)*, pp. 373–387, 2012. 53, 54

[63] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, May 2004. 56

[64] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and R. R, "The ucr time series classification/clustering homepage," *C.A.*, 2011. [Online]. Available: www.cs.ucr.edu/~eamonn/time_series_data/ 58

[65] E. Abreu, M. Lightstone, S. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," *Image Processing, IEEE Transactions on*, vol. 5, no. 6, pp. 1012–1025, 1996. 60

[66] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006. 62

[67] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991. 64

[68] D. B. Percival and A. T. Walden, *Wavelet methods for time series analysis.* Cambridge University Press, 2006, vol. 4. 64

[69] S. Papadimitriou and P. Yu, "Optimal multi-scale patterns in time series streams," in *Proceedings of the International Conference on Management of Data.* ACM, 2006, pp. 647–658. 64

[70] G. Batista, X. Wang, and E. J. Keogh, "A complexity-invariant distance measure for time series," in *SIAM Conference on Data Mining*, 2011. 67

[71] U. Hoffmann, G. Garcia, J. Vesin, K. Diserens, and T. Ebrahimi, "A Boosting Approach to P300 Detection with Application to Brain-Computer Interfaces," in *Proceedings of the IEEE EMBS Conference on Neural Engineering*. SPIE, 2005, pp. 97–100. 67

[72] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation MoCap database HDM05," Universität Bonn, Tech. Rep. CG-2007-2, June 2007. 70

[73] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR).* IEEE Computer Society, 2004, pp. 32–36. 73

[74] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, December 2007. 73

[75] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proceedings of the 14th International Conference on Computer Communications and Networks (ICCCN).* IEEE Computer Society, 2005, pp. 65–72. 73

[76] D. Waltisberg, A. Yao, J. Gall, and L. Van Gool, "Variations of a hough-voting action recognition system," in *Proceedings of the 20th International conference on Recognizing patterns in signals, speech, images, and videos.* Springer-Verlag, 2010, pp. 306–312. 75, 76

[77] M. S. Ryoo, C.-C. Chen, J. K. Aggarwal, and A. K. R. Chowdhury, "An overview of contest on semantic description of human activities (sdha) 2010," in *ICPR Contests*, 2010, pp. 270–285. 76

Vita

Shahriar Shariat Talkhoonche

Education

Spring 2009-Fall 2013 Ph. D. in Computer Science, Rutgers University

Fall 2005- Fall 2008 M. Sc. in Information Technology, Sharif University

Fall 2000- Spring 2005 B. Sc. in Computer Engineering, Isfahan University of Tech.

Experience

2009-2013 Teaching assistant, Department of Mathematics, Rutgers University, NJ

May 2011 - August 2011 Engineering Intern, Qualcomm Research, Bridgewater, NJ

June 2004 - December 2008 R&D Senior Software Engineer, Hamsoo IT Co. Esfahan, Iran

Publication

- Sh. Shariat, V. Pavlovic, "Improved sequence classification using adaptive segmental sequence alignment", Journal of Machine Learning W&CP vol 25 pp.379-394, 2012.

- Sh. Shariat, V. Pavlovic, "Isotonic CCA for Sequence Alignment and Activity Recognition", Published in Proc. of International Conference on Computer Vision (ICCV) 2011

- Sh. Shariat, V. Pavlovic, T. Papathomas,A. Braun, P. Sinha, "Sparse Dictionary Methods for EEG Signal Classification in Face Perception", Published in IEEE Workshop on Machine Learning in Signal Processing (MLSP) 2010.

- Sh. Shariat, M. Khansari, H. R. Rabiee, "Inferring a Bayesian Network for Content-base Image Classification", Communication in Computer and Information Scient, 2009, Vol 6. pp 211-218.

- Sh. Shariat, M. Khansari, H. R. Rabiee, "A New Measure for Precision Alternation in Merging Bayesian Networks in Image Classification Application", Published in proc. of 16th International Conference on Electrical Engineering 2008