Fast Approximate Dynamic Warping Kernels

G Nagendar and C.V. Jawahar
Center for Visual Information Technology, IIIT Hyderabad, India

ABSTRACT

The dynamic time warping (DTW) distance is a popular similarity measure for comparing time series data. It has been successfully applied in many fields like speech recognition, data mining and information retrieval to automatically cope with time deformations and variations in the length of the time dependent data. There have been attempts in the past to define kernels on DTW distance. These kernels try to approximate the DTW distance. However, these have quadratic complexity and these are computationally expensive for large time series. In this paper, we introduce FastDTW kernel, which is a linear approximation of the DTW kernel and can be used with linear SVM.

To compute the DTW distance for any given sequences, we need to find the optimal warping path from all the possible alignments, which is a computationally expensive operation. Instead of finding the optimal warping path for every pair of sequences, we learn a small set of global alignments from a given dataset and use these alignments for comparing the given sequences. In this work, we learn the principal global alignments for the given data by using the hidden structure of the alignments from the training data. Since we use only a small number of global alignments for comparing the given test sequences, our proposed approximation kernel is computationally efficient compared to previous kernels on DTW distance. Further, we also propose a approximate explicit featuremap for our proposed kernel. Our results show the efficiency of the proposed approximation kernel.

Keywords

DTW distance, Negative Definite Kernels, Kernel Approximation, Explicit Featuremaps

1. INTRODUCTION

Distance (Similarity) functions play an important role in a wide variety of problems, including regression, classification and clustering. They are used to find the similarity and dissimilarity between a pair of samples. There are several distance functions available in machine learning literature like Euclidean, Geodesic, Earth Movers Distance (EMD) [22], dynamic time warping (DTW) [2, 26], etc. Each of these distance functions computes different types of similarity, and is useful for different kinds of problems. For example, Euclidean distance is widely used in comparing the sequences which are in the same space. The main limitation of using Euclidean distance for time series data is that it does not capture the local dependencies between neighboring states of the time series, and is sensitive to distortion in time axis [13]. The best way to compare time series data is DTW distance [3, 26, 2]. For a given pair of time series, it finds the optimal alignment by ignoring both global and local shifts in the time dimension. DTW is often used in speech recognition [17] to determine the similarity between two speech signals representing the similar spoken words. In speech recognition, for a given same spoken word, the length of the signals are permitted to vary, but the overall speech signals are similar. In addition to speech recognition, DTW has also been found useful in many other disciplines [14], including word recognition [4, 21], bioinformatics [1], data mining and gesture recognition. DTW is commonly used in data mining as a distance measure between time series.

Over the last decade, SVMs has emerged as the most popular approach in classification. This is mainly due to its state-of-the-art performance on a wide variety of real-world classification problems [10, 12, 15]. Most of the distance functions may not yield positive definite kernels and thus can not be used along with SVM. Although these distance functions do not yield positive definite kernels, they have popular applications in many fields. For example, DTW distance does not yield a positive definite kernel but it works well on many time series classification problems with Nearest Neighbour (NN) classifier [30]. In general, SVM performs superior than NN on classification problems, where the appropriate choice of kernel exists [28]. Due to its superiority, it is reasonable to use DTW distance with SVM. However, because of its non-metric property, DTW distance does not yield positive definite kernel.

Non positive definite kernels are previously used in SVM [7, 29], and DTW distance previously used in kernel machines by many authors [2, 26, 32, 8]. In [8], the authors corrected the negative definite kernel matrix by its square. There are also some attempts where minor changes are incorporated in the definition of DTW distance for defining a positive definite kernel. Hayashi *et al.* [9] projected the time series into an Euclidean space such that the resulting representation approximates the DTW distance. Although, we can use the distance functions in kernels with some approximations/modifications in the original formulation, the resulting kernels are computationally expensive. This limits the use of these distance functions for large datasets. For example, the computational complexity for finding the DTW distance between two time series of length n and m respectively, is $O(nm)$. The resulting ker-

CODS '15, March 18 - 21, 2015, Bangalore, India

http://dx.doi.org/10.1145/2732587.2732592 ...$15.00.

nels over DTW distance has quadratic complexity, which is computationally prohibitive for large time series containing thousands of data points. Note that such large length time series commonly arises in many domains like speech, bioinformatics and text processing.

In this work, we propose the FastDTW kernel which is a linear approximation to the non-linear DTW kernel. Our main contributions are (i) computing the DTW distance based on a set of principal global alignments computed from the training data, and (ii) computing an approximate explicit feature map for the DTW kernel so that linear SVM can be used. The main objective of this paper is to speed up the DTW kernel by learning the optimal alignments from the given training data. As far as we are aware, none of the previous methods have exploited the hidden structure of the alignments. In our experiments, we observe that, we can represent the internal structure of the alignments using few basis alignments (global alignments). These basis alignments are sufficient for comparing any given new pair of sequences. Since we are avoiding computing the optimal alignments for the test sequences, the proposed FastDTW kernel is computationally efficient compared to previous techniques.

A popular method for speeding the non-linear kernels is with the help of explicit featuremap [27, 16, 20, 19]. Explicit featuremap approximates the original large dimensional featuremap by a small finite dimensional featuremap. This gives the linear approximation of the original non-linear kernel, which can then be used with a linear SVM. To find the linear approximation of the proposed Fast DTW kernel, we compute its approximate explicit featuremap using the global top alignments. Explicit featuremaps are quite popular for different non linear kernels like intersection [16], Hellinger's, χ^2 and RBF kernels [27, 20]. However, this technique does not appear to be in widespread use in the time series community. In this paper, we follow this line of work and propose a novel approximate explicit feature map for DTW kernel. In [25], the authors proposed a dynamic time warping algorithm for the computation of DTW distance, which is linear in both time and space. The algorithm finds a nearly optimal warp path between two time series. However, the limitation of this algorithm is that it cannot be transformed into valid kernel. For a given dataset, we first learn the global top alignments, and then compute the explicit featuremap for the DTW kernel from these alignments. Our proposed approximation kernel is computationally efficient compared to other kernels over DTW distance. We compare our proposed kernel with the GA kernel [5] and, Gaussian DTW kernel [2], and show speed up for the given datasets. Since we are approximating the DTW kernel, the FastDTW kernel slightly decreases accuracy.

The paper is organized as follows. The next section describes the popular dynamic time warping technique and some existing kernels based on DTW distance. Details about problem formulation are discussed in Section 3. Section 4 provides a detailed explanation of our proposed approximation techniques. Section 5 discusses experimental evaluations of proposed kernel with previous kernels on DTW distance, followed by concluding remarks.

2. PRELIMINARIES

Kernel methods have been successfully used for comparing images, graphs and strings. Defining appropriate kernels for comparing structured objects like time series, remains a key challenge in the application of kernel methods to areas like signal processing, speech recognition and datamining.

We cannot directly use popular kernels such as Gaussian and polynomial kernels for comparing time series. This is mainly due to the variable length of time series. Also these kernels cannot capture the local dependencies between neighboring states of the time series. A successful method to compare time series is dynamic time warping (DTW) [24, 26, 2]. For the given two time series, the DTW yields an optimal alignment from all the possible alignments. This optimal alignment can be used to find the corresponding regions between the given time series. It can also be used to find the similarity between the time series. The DTW measure is the difference between the two time series after they have been mapped together with the optimal alignment.

Given two time series, $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_m)$ of lengths n and m respectively, an alignment π of length $|\pi| = p$ is a pair of increasing p-tuples (π_1, π_2) such that

$$1 = \pi_1(1) \leq \ldots \leq \pi_1(p) = n,$$
$$1 = \pi_2(1) \leq \ldots \leq \pi_2(p) = m$$

with unitary increments and no simultaneous repetitions. That is, for $\forall 1 \leq i \leq p-1$,

$$\pi_1(i+1) \leq \pi_1(i) + 1, \quad \pi_2(i+1) \leq \pi_2(i) + 1$$
$$(\pi_1(i+1) - \pi_1(i)) + (\pi_2(i+1) - \pi_2(i)) \geq 1.$$

Intuitively, an alignment π between X and Y describes a way to associate each element of X to one or possibly more elements in Y, and vice-versa. Let $A(X, Y)$ be the set of all possible alignments between X and Y. The DTW distance between two time series (sequences) X and Y is given as the minimum distance over all possible alignments,

$$DTW(X, Y) = min_{\pi \in A(X,Y)} \sum_{i=1}^{|\pi|} \varphi(X_{\pi_1(i)}, Y_{\pi_2(i)})$$

where, $\varphi(X_{\pi_1(i)}, Y_{\pi_2(i)})$ is the distance between the given sequences indexed by the alignment π. In general, Euclidean distance is used for this measure. The resulting alignment is the optimal path between the given two sequences.

For comparing two time series X and Y of length n and m respectively, the time and space complexity of DTW is $O(nm)$. The quadratic complexity mainly involves finding the optimal alignment from a huge set of possible alignments. The quadratic complexity is particularly prohibitive for large length time series. For the time series containing few thousands of measurements, it requires lot of memory and is computationally unattractive. In practice, heuristic constraints are used to speed up DTW. Among these, two of the most commonly used constraints are Sakoe-Chiba [23] and Itakura [11] constraints. Instead of searching in the entire space for optimal path, these restrict the search space using a constraint window. When constraints are used, the DTW algorithm finds the optimal path which passes only through the constraint window. It speeds up the DTW computation by a constant factor but the quadratic complexity still remains. In addition, using these methods the global optimal path can not be found if it does not pass through the constraint window and constraints work well if the optimal path is expected to be closer to the diagonal.

Since SVMs often outperforms K-NN classifiers [28], it is desirable to use DTW distance with SVMs. To do this, we need to define a valid kernel using DTW distance. Several kernels have been proposed over the DTW distance [2, 26, 32, 8]. For example, in [2], the authors proposed the Gaussian DTW (GDTW) kernel based on DTW distance as follows

$$\kappa_{GDTW} = exp\Big(\underset{\pi \in A(X,Y)}{-min} \frac{1}{|\pi|} \sum_{i=1}^{|\pi|} \|X_{\pi_1(i)} - Y_{\pi_2(i)}\|_2^2 \Big) \quad (1)$$

Here Euclidean distance is used for the distance measure φ. By using Gaussian kernel for φ, in [26], the authors defined another kernel over DTW, which is given as

$$\kappa_{DTW_1} = \max_{\pi \in A(X,Y)} \frac{1}{|\pi|} \sum_{i=1}^{|\pi|} e^{\frac{-1}{\sigma^2}\|X_{\pi_1(i)} - Y_{\pi_2(i)}\|_2^2} \quad (2)$$

To get the better performance using DTW kernel, Cuturi *et al.* [6] proposed Group Alignment (GA) kernel. Their kernel is not based on the optimal alignment, but takes advantage of all the scores obtained from all the possible alignments. This kernel is defined as,

$$\kappa_{GA}(X,Y) = \sum_{\pi \in A(X,Y)} \prod_{i=1}^{|\pi|} \kappa(X_{\pi_1(i)}, Y_{\pi_2(i)}) \quad (3)$$

where, $\kappa(X_i, Y_j) = e^{-\varphi(X_i, Y_j)}$. This kernel is positive definite if $\frac{\kappa}{\kappa+1}$ is positive definite. Rather than considering the optimal alignment (simple minimum of the objective function), the kernel considers softmax of the scores of all possible alignments. Intuitively, the kernel considers both the optimal alignment and all the alignments which are closer to it. They argue that, two sequences are similar not only if they have one single alignment with high score, but also share a wide set of other alignments. All the above kernels have quadratic complexity. This is computationally expensive for large datasets containing thousands of sequences with length in thousands. Based on the constraint windows [18] for DTW distance, Triangular Global Alignment (TGA) is proposed in [5] for speeding up GA kernel. This increase in speed comes at the cost of an approximation, and the resulting path may be suboptimal. This reduces the computational cost by a constant factor. In all the above kernels, the quadratic complexity mainly involves finding the optimal alignment or near optimal alignments.

GA kernel considers all the alignments or alignments near to the diagonal, due to which it is better than DTW kernel. Since, it needs to compute the alignments for every pair of sequences it is computationally slower. In this paper, instead of finding the optimal alignments for the given time series, we learn a set of global alignments from the training data and use these alignments for comparing new pair of sequences. Since we are finding the global alignments from all the possible alignments, in one way, our proposed kernel is an approximation to the GA kernel.

In general, SVMs with non linear kernels over large data sets are computationally slower compared to linear kernels. It requires efficient solvers to optimize the given problem. A linear SVM is given by the inner product $F(X) = \langle w, X \rangle$ between the data sample X and a weight vector w. On the other hand, a non linear SVM is given by the expansion $F(X) = \sum_{i=1}^{M} \alpha_i \kappa(X, X_i)$, where κ is a non-linear kernel, X_is are representative data vectors and M is the number of support vectors. In most of the cases, evaluating the inner product $\langle w, X \rangle$ is more efficient than evaluating the kernel $\kappa(X, X_i)$. This makes the linear SVM atleast M times faster compared to non-linear SVM, which is a significant gain especially on large datasets. The training time is also effected in the similar way. A successful way to speed up the non-linear SVM is with the help of explicit featuremaps [27]. For a given non-linear kernel κ, there exists a featuremap ϕ such that $\kappa(X,Y) = \langle \phi(X), \phi(Y) \rangle$. However, in most of the cases, the featuremap ϕ is of infinite dimension. In explicit featuremap, it finds a finite dimension approximation $\phi^{'}$ of the featuremap ϕ such that $\kappa(X,Y) \simeq \langle \phi^{'}(X), \phi^{'}(Y) \rangle$. Given a positive definite kernel $\kappa(X,Y)$ and a data density $p(\mathcal{X})$, where $X, Y \in \mathcal{X}^D$ for some input space $\mathcal{X}$, the approximation finds the featuremap $\phi^{'}$, which minimizes the following functional

$$E(\phi^{'}) = \int_{\mathcal{X}^D \times \mathcal{X}^D} (\kappa(X,Y) - \langle \phi^{'}(X), \phi^{'}(Y) \rangle)^2 p(X) p(Y) dX dY$$

where, the components $\phi_k^{'}(X)$ $(k = 1, 2, \ldots)$ are eigenfunctions of the kernel κ. In general, the data density $p(\mathcal{X})$ is approximated by a finite sample set and correspondingly eigenfunctions are replaced with eigenvectors. The eigenfunctions (vectors) play an important role in the approximation of the kernels.

3. PROBLEM FORMULATION

The problem of finding optimal alignment in DTW is solved as follows: For given pair of two time series $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$, where $x_i, y_i \in \mathbb{R}^k$, of length n and m respectively, we first construct a cost matrix C of dimension $n \times m$. The $(i,j)^{th}$ element of the cost matrix C is the Euclidean distance between the elements x_i and y_j. The x-axis and y-axis of the matrix C represent time domain for the time series X and Y respectively. Each warp path in the cost matrix represents an alignment between the time series X and Y, and the sum of elements through the alignment gives the cost of the alignment. The DTW finds an optimal warp path (top alignment) which has the minimum cost. We need to compute the optimal warp path for finding the DTW distance between two time series.

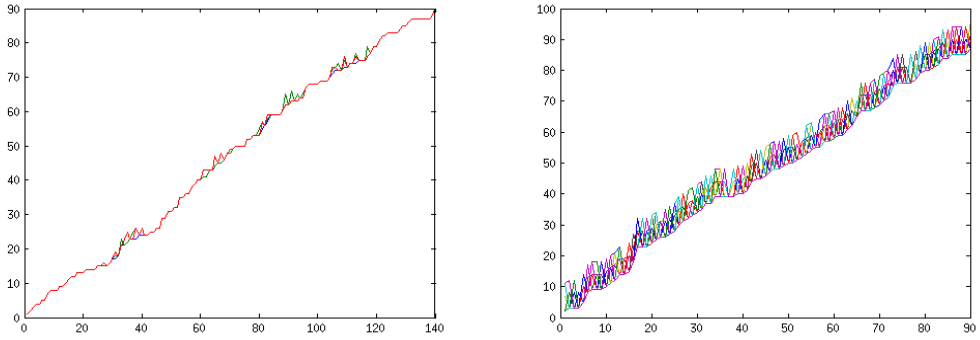
It is likely that for given two distinct pairs of time series, there exists some common top alignments. For a given dataset, the top alignments may not be completely different for two different class of time series. The top alignments between any two time series from two different classes always follow some structure, which depends on the dataset. We plot the top 3 alignments between two time series in Figure 1(a). The top alignments are not completely different from each other. We show this behavior over a subset of samples from Libra dataset in Figure 1(b). Here, we plot the top alignments between all the pairs of samples. Clearly, we can observe that all the alignments are passing through a small window. Any of the top alignments is not completely different from others. Since for any given two time series, their top alignments are following some structure, we can approximate the top alignments for a new pair of time series from the few alignments obtained from the given data. The aim of this work is to find these few candidate top alignments (global alignments) from the given data.

Let us define the DTW kernel as follows

$$\kappa_{DTW} = exp\Big(- \sum_{\pi \in A(X,Y)} \frac{1}{|\pi|} \sum_{i=1}^{|\pi|} \|X_{\pi_1(i)} - Y_{\pi_2(i)}\|_2^2 \Big) \quad (4)$$

4. APPROXIMATION FOR DTW KERNELS

In general, explicit featuremaps for non-linear kernels are computed using their eigenfunctions. Eigenfunctions play an important role in representing any kernel function and, from Mercer theorem, we can represent a kernel function using sum of its eigenfunctions. However, computing eigenfunctions for a kernel which is a solution to an optimization problem is practically difficult. In DTW distance, alignments have some similar properties as eigenfunction. For a given pair of samples, we can represent its DTW distance using few alignments, however, these alignments will change across the samples. Since alignments play a similar role as eigenfunctions in DTW distance, we compute the global top alignments (eigen alignments) for the given data. These alignments are good enough for comparing any samples from the given data. In this paper, we introduce two methods for finding the global top alignments from



(a) (b)

Figure 1: (a) The top 3 alignments between two time series X and Y of length 90 and 140 respectively. These alignments have least cost compared to other alignments. These alignments are not completely different from each other. (b) Optimal warp paths between the sequences of a subset of Libra dataset.

the given data. In the first method, we compute the global top alignments from the total top alignments of the given data. For a given pair of samples, their top alignments are computed from their cost matrix. Since there are similarities between the top alignments for different pairs of samples, there will be similarities between their cost matrices also. Instead of computing representative alignments from the total top alignments, in the second method, we compute the representative cost matrix for the given data and global top alignments are computed from this cost matrix.

4.1 Approximation using Top Alignments (ATA)

To model the global top alignments, we use principal component analysis (PCA). The objective is to find the global top alignments from the optimal alignments obtained from the given data. We need to find the better representative alignments from the total optimal alignments. The problem is similar to the objective of PCA, where for a given data it finds the principle directions in which variance of the data is maximum and the data can be well represented using these representations. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of correlated observations into a set of values of linearly uncorrelated variables called principal components. The number of principal components are less than or equal to the number of original variables. This transformation computes the principal components in such a way that the first principal component has the largest possible variance and accounts for as much of the variability in the data as possible. The succeeding components in turn has the next highest variance possible. This method is very much similar to the objective of the approximation using top alignments, where we find the representative alignments (principal alignments) from the total alignments obtained from the given data. However, PCA works only on one dimensional data (1D), if we want to apply over two dimensional (2D) data, the 2D data must be transformed into 1D data. This leads to a high dimensional vector space where it becomes difficult to evaluate the covariance matrix accurately. Similar to PCA, a two-dimensional principal component analysis (2DPCA) [31] is proposed to overcome these issues. 2DPCA is based on 2D matrices rather than 1D vectors. In this work, we use 2DPCA for computing the global top alignments from the total optimal alignments obtained from the given data. The global top alignments are computed as follows. First we represent each alignment using a 2D matrix. The global top alignments are then computed by applying 2DPCA over these matrices.

An alignment π between two time series of length n and m is represented using a $n \times m$ grid. Equivalently, it can also be represented using a $n \times m$ binary matrix (alignment matrix), where the elements are either 0 or 1. This representation is given in Figure 2. The entries in the matrix through which the alignment passes is 1 and for other entries it is 0. Since for every pair of time series there exist many alignments, it corresponds to many binary matrices. For a given pair of time series X and Y, denote their possible alignments as $\pi_1, \pi_2, \ldots, \pi_l$, where l is the total possible number of alignments. Assume that these alignments are arranged according to their cost, i.e π_1 is the optimal alignment which has the least cost and π_k is the top k^{th} alignment for X and Y. For the time series X and Y, we represent its final alignment matrix as follows

$$B_{X,Y} = \cup_{k=1}^{l} B_{X,Y}^{k}$$

where, $B_{X,Y}^{k}$ is the alignment matrix for the time series X and Y corresponding to the k^{th} alignment, and l is the total number of possible alignments.

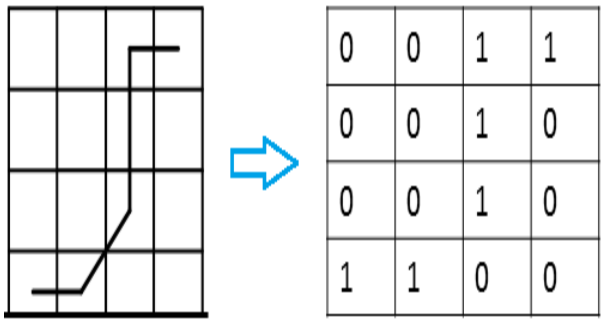


Figure 2: Alignment matrix representation. The entries in the matrix where the alignment passes is 1, and for others entries it is 0.

For a given dataset, we first construct these alignment matrices for every pair of samples. Since there exist possibly many alignments for every pair of samples and only top alignments have significance in DTW distance, we take only top t alignments. For a given dataset $\mathcal{X}$, we construct its alignment matrices as follows,

$$P_t = \cup_{X,Y \in \mathcal{X}} (\cup_{j=1}^{t} B_{X,Y}^{j})$$

Here, the set P_t takes only top t alignments between every pair of samples. Now, the set P_t contains all the best possible alignment matrices for the given dataset. We compute the global top representative alignments from this set of alignment matrices using 2DPCA. If the dataset contains alignments of variable length, the set P_t contains matrices of variable dimension. In this case, we cannot apply the above procedure. To overcome this, we first scale the alignments to a fixed size and, then alignment matrices are computed from these scaled alignments. The resulting alignment matrices will thus have the same dimension. We then apply 2DPCA over this set of matrices and find the eigenvectors. These eigenvectors give the global top alignments for the given data. The resulting kernel is given as follows,

$$\kappa_{ATA}(X,Y) = \sum_{\pi \in G_{\mathcal{X}}} e^{-\beta \sum_{k=1}^{|\pi|} (X_{\pi(k)} - Y_{\pi(k)})^2} \tag{5}$$

where $G_{\mathcal{X}}$ is the set of global top alignments of $\mathcal{X}$.

Now our objective is to find the explicit featuremap for this kernel. The kernel given in (5) is the sum of RBF kernels over the global top alignments. From the work on explicit featuremap for RBF kernel [27], the explicit featuremap of dimension n for RBF kernel is given as follows

$$\phi_{RBF}(X) = \frac{1}{\sqrt{n}} [e^{-i\langle \omega_1, X \rangle}, \ldots, e^{-i\langle \omega_n, X \rangle}] \tag{6}$$

where $\omega_1, \ldots, \omega_n$ are sampled from the Gaussian density. Since the kernel given in Equation 5 is the sum of Gaussian kernels over the global top alignments, the final explicit featuremap for this kernel is given as follows,

$$\phi_{ATA}(X) = \frac{1}{\sqrt{n}} [e^{-i\langle \omega_1, X_{\pi_1} \rangle}, \ldots, e^{-i\langle \omega_n, X_{\pi_1} \rangle}, \ldots, e^{-i\langle \omega_1, X_{\pi_m} \rangle}, \ldots, e^{-i\langle \omega_n, X_{\pi_m} \rangle}] \tag{7}$$

where $\pi_1, \ldots, \pi_m$ are the global top alignments and m is the total number of global top alignments. The proposed linear approximation of the DTW kernel is given as

$$\kappa_{ATA}^{lin}(X,Y) = \langle \phi_{ATA}(X), \phi_{ATA}(Y) \rangle \tag{8}$$

4.2 Approximation using Cost Matrix (ACM)

In the approximation using top alignments (ATA), to compute the global top alignments, we need to compute the top alignments for every pair of samples. This is computationally costly for large datasets. To avoid this, we propose another technique, in which a global cost matrix is computed from the given data and the global top alignments are computed from this cost matrix.

To find the DTW distance between two time series, we first need to compute the cost matrix. The optimal alignments are computed from this cost matrix. In the approximation using top alignments, we construct the cost for every pair of samples, and top alignments from these cost matrices are computed for learning the global top alignments for the given data. These cost matrices vary across the samples. Learning the global top alignments from the top alignments obtained from the cost matrices is computationally expensive for large datasets. Since there are similarities between the optimal alignments between different pairs of sequences, there would exists some similarities between the their cost matrices also. The idea is to incorporate all these similarities into a single cost matrix called global cost matrix, which captures all the correlations present in the data. Now, the global top alignments are computed from this global cost matrix. We construct the global cost matrix for the given data as follows. We first consider cost matrix for every pair of time series from the given data, and normalize each cost matrix by the maximum cost of that matrix, then we take mean of all these matrices as the global cost matrix for the given data. If the dataset contains alignments of variable length, the cost matrices will have variable dimension. In this case, we can not compute the global cost matrix as discussed above. To overcome this, we first scale the alignments to a fixed size, and then cost matrices are computed from these scaled alignments. The resulting cost matrices will thus have the same dimension. The top alignments computed from the global cost matrix give the global top alignments for the given data. These alignments are sufficient enough for comparing any two time series. Since, we have global top alignments, we compute the explicit featuremap as described in method ATA. The DTW kernel obtained from this global top alignments is given as,

$$\kappa_{ACM}(X,Y) = \sum_{\pi \in G'_{\mathcal{X}}} e^{-\beta \sum_{k=1}^{|\pi|} (X_{\pi(k)} - Y_{\pi(k)})^2} \tag{9}$$

where $G'_{\mathcal{X}}$ is the set of global top alignments computed from the global cost matrix. The explicit featuremap for this method is given as follows

$$\phi_{ACM}(X) = \frac{1}{\sqrt{n}} [e^{-i\langle \omega_1, X_{\pi'_1} \rangle}, \ldots, e^{-i\langle \omega_n, X_{\pi'_1} \rangle}, \ldots, e^{-i\langle \omega_1, X_{\pi'_m} \rangle}, \ldots, e^{-i\langle \omega_n, X_{\pi'_m} \rangle}] \tag{10}$$

where $\pi'_1, \ldots, \pi'_m$ are the global top alignments and m is the total number of global top alignments.

Since, we are avoiding learning the global top alignments from a large set of top alignments, ACM is computationally faster compared to the approximation ATA. However, as we are computing the global cost matrix from a large set of cost matrices, it may not capture all the correlations present in the data. Due to this, the proposed ATA has slightly reduced accuracy. We refer the kernels obtained from the two approximation techniques ATA and ACM as FastDTW kernels.

5. EXPERIMENTS

5.1 Datasets

The goal of this evaluation is to demonstrate the efficiency of the proposed kernel on a wide range of time series data sets. We evaluate our kernel over popular machine learning datasets Libra, Auslan, Japanese vowels, handwritten Characters and PEMS database of freeway traffic. Except Libra dataset, all the other datasets contains multivariate time series of variable length. In addition to these small datasets, we also evaluate our FastDTW kernel on large scale time series datasets obtained from UCR Time Series Data Mining Archive. Since our method is applicable only for fixed length sequences, we scale the variable length sequences to a fixed length using standard scaling techniques. For all these datasets, we com-

Database	dimension	length	classes	# samples
Libra	2	45	15	945
Auslan	22	45-136	95	2465
JV	12	7-29	9	640
HC	3	60-182	20	2858
PEMS	963	144	7	440

Table 1: Details of the datasets considered in the experiments Libra, Auslan, Japanese Vowels (JV) and Handwritten Characters (HC). Here, Libra dataset contains fixed length time series, whereas all other datasets describes multivariate time series.

	Libra	Auslan	PEMS	HC	JV
# Samples	945	2465	440	2858	640
# Global Alignments	14	24	8	24	12

Table 2: Number of global alignments for the datasets used in the experiments. Here, the number of global alignments are based on the size of a dataset.

pare our methods with (GDTW) [2] kernel and (GA) Kernel [5]. Details of the datasets are given in Table 1.

5.2 Experimental results

5.2.1 Procedure and Evaluation

We measure the efficiency of the proposed approximations ATA and ACM in time with respect to the number of samples, and compared to the GA [5] kernel and GDTW [2] kernel. In our proposed approximation kernels, we use FastDTW [25] implementation for computing both top alignments and cost matrix. For every pair of time series, we take $t = 10$, i.e we choose top 10 alignments. In both the proposed approximations, the number of global alignments are based on the size of the dataset. The number of global alignments for the given datasets are shown in Table 2. All the kernels are implemented in Matlab. All experiments are carried out on a single core of a 2.1 GHz AMD 6172 processor with 12 Gb RAM. For comparison, the runtime is measured using the system clock with minimal background processes running. In both the proposed approximations ATA and ACM, we consider equal number of global top alignments. Hence the runtime for kernel computation is same for both the methods. As both the approximation kernels have equal runtime, in all our experiments, for comparing computational time with other kernels, we refer our proposed approximation kernel as FastDTW kernel.

5.2.2 Role of number of Global Alignments

Since we select the number of global alignments based on the size of the dataset, we show the effect of the number of global alignments on accuracy in Table 3. To show the role of number of global alignments, we compare proposed ATA over varying number of selected global alignments with the GA kernel and GDTW kernel. As we increase the number of top alignments, the accuracy improves. This suggests that to get better accuracy we need to consider more global alignments. The results also shows that, if we further increase the number of alignments, there is no improvement in the accuracy.

5.2.3 Accuracy of FastDTW Kernel

We compare the accuracy of proposed approximation methods ATA and ACM with GA kernel and GDTW kernel on the given datasets in Figure 3(a). On Libra dataset, our proposed methods achieve better accuracy compared to GDTW kernel, and performing comparable to GA kernel. This shows that our proposed kernel performs equally well, if not better than GA kernel on same length time series. For other datasets, our results are superior compared to GDTW kernel and comparable to the GA kernel. The drop in the performance for variable length time series is due to the scaling step which we are performing for scaling the data to equal length. While scaling the data to equal size, we are loosing some information from the data, which causes the degraded performance. However, compared to GDTW kernel, which is exponential of DTW distance, the proposed methods get better accuracy over all the datasets.

	Libra	Auslan	PEMS	HC	JV
ATA	12.2±0.29	43.1±2.3	25.2±0.8	45.2±2.8	17.1±0.3
ACM	**7.7±0.21**	**28.9±1.2**	**19.5±0.3**	**36.3±1.8**	**12.3±0.2**

Table 5: Training time (Hrs) for ATA and ACM over the given datasets. Only top 1 alignment is considered in this experiment.

Sometimes, the alignments may not follow the unimodal distribution. In such case, if we apply PCA directly over the alignments, we may not get the correct candidate global alignments. Due to this, the resulting global alignments may reduce the accuracy. To get better model for the alignments, we use bimodal distribution over the considered alignments. In bimodal distribution, we apply the PCA over each distribution separately, and global alignments from these distributions are taken as the final global alignments. The results of these experiments are shown in Table 4. Clearly, we obtain better performance using bimodal compared to unimodal distribution. This suggests that if data is well distributed or follows some distribution, we get better global alignments.

5.2.4 Efficiency

In this section, we explore the efficiency of FastDTW kernel over small and large datasets. In all the experiments, we compare FastDTW kernel with GA kernel and GDTW [2] kernel. In the approximation using top alignments (ATA), we need to compute the top alignments for every pair of time series. Due to this ATA is computationally slower compared to ACM in training. Training time for both the approximations ATA and ACM over the given datasets are given in Table 5. It shows that the approximation ACM is computationally faster compared to ATA over all the datasets. We compare our proposed FastDTW kernel with the GA kernel and GDTW kernel over all the datasets and results are shown in Figure 3(b). The results show that FastDTW kernel is computationally faster compared to both the GA kernel and GDTW kernel over all the datasets. For Libra dataset, which is the smallest dataset in our experiments, both FastDTW kernel and GA kernel are performing equally well. However, for other datasets, which have more number of sequences compared to Libra, FastDTW kernel has significant gain in performance compared to GA kernel. Also the proposed FastDTW kernel clearly outperforms GDTW kernel over all the datasets. On Hand written characters dataset, FastDTW kernel is nearly 10 times faster than GDTW kernel. The speed up of our proposed kernel is mainly due to the global top alignments, which are then used in the computation of explicit featuremap. To further explore the FastDTW kernel over large length sequences, we test the proposed kernel over UCR time series data mining archive. This dataset contains time series of length from 1000 to 0.1 Million. In Figure 4, we show the computational time over varying length of time series. For small length sequences, both FastDTW kernel and GA kernel are performing equally well. The performance of FastDTW kernel is least effected with the increase

	Libras		Auslan		PEMS		HC		JV	
	Accuracy	Test time	Accuracy	Test time	Accuracy	Test time	Accuracy	Test time	Accuracy	Test time
Top 1 Alignment	82.2	**0.9**	83.8	**2.3**	71.0	**2.2**	86.2	**3.1**	84.9	**2.4**
Top 3 Alignments	82.5	**0.9**	84.1	**2.3**	71.5	**2.2**	86.1	**3.1**	85.5	**2.4**
Top 5 Alignments	83.1	**0.9**	84.4	**2.3**	71.8	**2.2**	86.3	**3.1**	85.6	**2.4**
Top 8 Alignments	83.4	**0.9**	85.0	**2.3**	72.1	**2.2**	86.7	**3.1**	85.8	**2.4**
Top 15 Alignments	83.5	**0.9**	85.1	**2.3**	72.1	**2.2**	87.0	**3.1**	86.1	**2.4**
GDTW Kernel	83.4	2.4	84.6	13.2	71.6	14.7	86.1	29.3	85.4	13.2
GA Kernel	**83.6**	1.1	**87.3**	5.1	**73.8**	5.2	**89.3**	11.2	**88.7**	4.8

Table 3: Performance of proposed ATA over varying number of global alignments. Here, both the accuracy (%) and test time is compared with GA kernel. Proposed FastDTW kernel is computationally faster with a slight decrease in the accuracy

	Unimodal					Bimodal				
	Libra	Auslan	PEMS	HC	JV	Libra	Auslan	PEMS	HC	JV
ATA	83.5±0.9	85.3±1.0	72.1±0.4	87.1±1.3	86.1±0.7	83.6±0.8	86.9±1.2	72.5±0.5	87.4±1.5	87.1±0.9
ACM	83.5±0.8	84.5±0.9	71.4±0.3	86.2±1.0	84.9±0.6	83.6±0.7	86.2±1.1	72.1±0.5	87±0.9	86.9±0.9

Table 4: Comparison between Unimodal and Bimodal distributions over Libra, Auslan, PEMS, HC (Hand written Characters) and JV (Japanese Vowels) datasets. GA kernel does not uses Bimodal distribution.

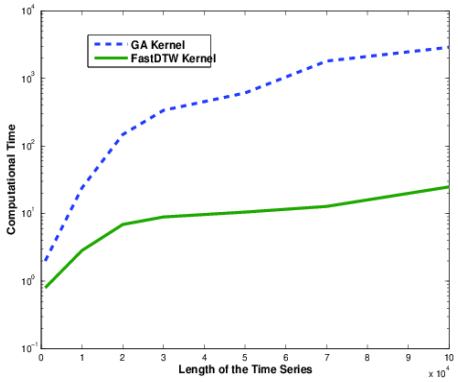


Figure 4: Comparison of FastDTW kernel with GA kernel over varying length of time series. Here, the computational time is given in minutes and is shown on log scale. For small number of samples both FastDTW kernel and GA kernel are performing equally well. The performance of FastDTW kernel has minimal effect with the increase in the number of samples.

in the number of samples. On the other hand, in case of GA kernel, test time increases significantly with increase in the number of samples. For time series of length 0.1 Million, FastDTW kernel is able to compute in around one minute, whereas for GA kernel, it took nearly one hour. For better comparison, we show the computational time on log scale. This suggests that proposed FastDTW kernel is computationally efficient compared to GA kernel and is suitable for both large datasets and large length time series.

6. CONCLUSION

In this paper, we propose the linear approximation to the non-linear DTW kernel. In addition, we also propose the explicit featuremap for our proposed approximation kernel. For computing the explicit featuremap, we exploit the internal hidden structure of the alignments. We represent the optimal alignments between the time series using few global top alignments. This reduces the computational cost of DTW distance by huge factor. The main aim of this paper is to explore the hidden structure of the alignments for approximating the non linear DTW kernel by a linear kernel.

References

[1] John Aach and George M. Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6):495–508, 2001.

[2] Claus Bahlmann, Bernard Haasdonk, and Hans Burkhardt. On-line handwriting recognition with support vector machines - a kernel approach. In *Proc. of the 8th IWFHR*, pages 49–54, 2002.

[3] Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop, Seattle, Washington, July 1994. Technical Report WS-94-03*, pages 359–370, 1994.

[4] Michael K. Brown and Lawrence R. Rabiner. Dynamic time warping for isolated word recognition based on ordered graph searching techniques. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '82, Paris, France, May 3-5, 1982*, pages 1255–1258, 1982.

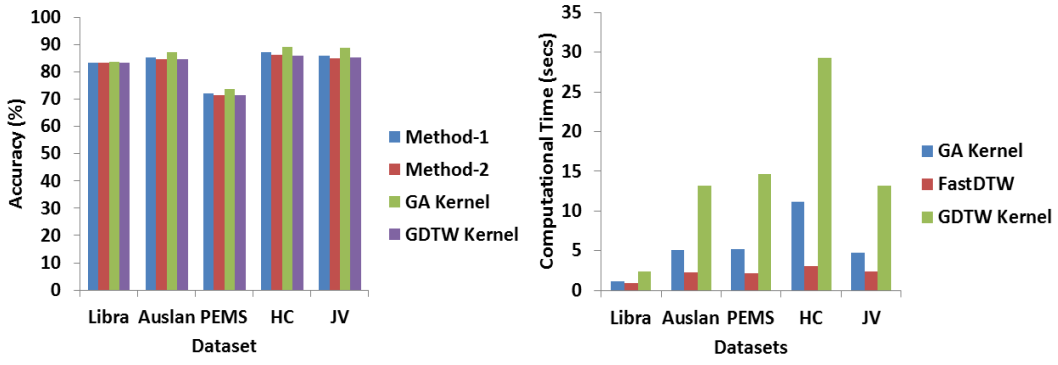
[5] Marco Cuturi. Fast global alignment kernels. In *ICML*, pages 929–936, 2011.

[6] Marco Cuturi, Jean-Philippe Vert, Oystein Birkenes, and Tomoko Matsui. A kernel for time series based on global alignments. *CoRR*, abs/cs/0610033, 2006.

[7] Dennis DeCoste and Bernhard Schölkopf. Training invariant support vector machines. *Machine Learning*, 46(1-3):161–190, 2002.

[8] Steinn Gudmundsson, Thomas Philip Runarsson, and Sven Sigurdsson. Support vector machines and dynamic time warping for time series. In *IJCNN*, pages 2772–2776. IEEE, 2008.

[9] Akira Hayashi, Yuko Mizuhara, and Nobuo Suematsu. Embedding time series data for classification. In *MLDM*, pages 356–365, 2005.



(a) (b)

Figure 3: (a) Performance of the proposed approximation methods ATA and ACM with the GA kernel over Libra, Auslan, PEMS, HC (Hand written Characters) and JV (Japanese Vowels) datasets. (b) Since the computational time for both the methods ATA and ACM is same, we are referring them Fast DTW kernel. Comparison of proposed FastDTW kernel with GA kernel over the given datasets. Clearly, FastDTW kernel is computationally faster compared to GA kernel. Here, computational time is given seconds.

[10] Hugo Hidalgo, Sonia Sosa León, and Enrique Gómez-Treviño. Application of the kernel method to the inverse geosounding problem. *Neural Networks*, 16(3-4):349–353, 2003.

[11] Fumitada Itakura. Readings in speech recognition. chapter Minimum Prediction Residual Principle Applied to Speech Recognition, pages 154–158. 1990.

[12] Yuh jye Lee and O. L. Mangasarian. Ssvm: A smooth support vector machine for classification. Technical report, Computational Optimization and Applications, 1999.

[13] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, March 2005.

[14] Eamonn J. Keogh and Michael J. Pazzani. Derivative dynamic time warping. In *In First SIAM International Conference on Data Mining (SDMâĂŹ2001*, 2001.

[15] Kwang In Kim, Keechul Jung, Se Hyun Park, and Hang Joon Kim. Support vector machines for texture classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(11):1542–1550, November 2002.

[16] Subhransu Maji and Alexander C. Berg. Max-margin additive classifiers for detection. In *ICCV*, pages 40–47, 2009.

[17] Cory S. Myers, Lawrence R. Rabiner, and Aaron E. Rosenberg. An investigation of the use of dynamic time warping for word spotting and connected speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '80, Denver, Colorado, April 9-11, 1980*, pages 173–177, 1980.

[18] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., 1993.

[19] Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *NIPS*, pages 1509–1517, 2009.

[20] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, 2007.

[21] Viresh Ranjan, Gaurav Harit, and C.V Jawahar. Document retrieval with unlimited vocabulary. In *IEEE Winter Conference on Applications of Computer Vision*, 2015.

[22] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *ICCV*, pages 59–66, 1998.

[23] Hiroaki Sakoe. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26:43–49, 1978.

[24] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 43–49, 1978.

[25] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, 2007.

[26] Hiroshi Shimodaira, Ken ichi Noma, Mitsuru Nakai, and Shigeki Sagayama. Dynamic time-alignment kernel in support vector machine. In *NIPS*, pages 921–928, 2001.

[27] Andrea Vedaldi and Andrew Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(3):480–492, 2012.

[28] Pascal Vincent and Yoshua Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. Technical Report 1197, Département d'informatique et recherche opérationnelle, Université de Montréal, 2001.

[29] Christian Wallraven, Barbara Caputo, and Arnulf B. A. Graf. Recognition with local features: the kernel recipe. In *ICCV*, pages 257–264, 2003.

[30] Xiaopeng Xi, Eamonn J. Keogh, Christian R. Shelton, Li Wei, and Chotirat Ann Ratanamahatana. Fast time series classification using numerosity reduction. In *ICML*, pages 1033–1040, 2006.

[31] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-Yu Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(1):131–137, 2004.

[32] Feng Zhou, Fernando De la Torre, and Jeffrey F. Cohn. Unsupervised discovery of facial events. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.