A Study on the Dynamic Time Warping in Kernel Machines

Hansheng Lei
Computer Science Department
The University of Texas at Brownsville
80 Fort Brown, Brownsville, TX 78520, USA
hansheng.lei@utb.edu

Bingyu Sun
Institute of Intelligence Machine
Chinese Academy of Science
Hefei, Anhui, 230031, PR. China
bysun@ustc.edu

Abstract

The Dynamic Time Warping (DTW) is state-of-the-art distance measure widely used in sequential pattern matching and it outperforms Euclidean distance in most cases because its matching is elastic and robust. It is tempting to substitute DTW distance for Euclidean distance in the Gaussian RBF kernel and plug it into the state-of-the-art classifier Support Vector Machines (SVMs) for sequence classification. However, it is not straightforward that DTW also outperforms Euclidean distance in kernel machines. While counter-examples can be found to numerically prove that DTW is not Positive Definite Symmetric (PDS) acceptable by SVM, little is known why it can not be PDS theoretically. We analyze the DTW kernel and complete a theoretical proof via the connection between PDS kernel and Reproducing Kernel Hilbert Space (RKHS). Our analysis leads to a better understanding that all Hilbertian metrics can be be converted to a PDS kernel in the Gaussian form, while the reverse is not true. The proof can be extended to conclude that elastic matching distance is not eligible to construct PDS kernels (e.g., Edit distance). Experiments were conducted to compare the RBF-kernel and DTW kernel in SVM classifications and the results show that simple Euclidean distance outperforms DTW in kernel machines.

1 Introduction

Sequential patterns are common with broad applications in online handwriting recognition, speech recognition and streaming data analysis. There are a large number of distance measures proposed from sequential pattern matching [12, 13, 14]. Comprehensive applications have shown that the Dynamic Time Warping (DTW) and even the simple Euclidean distance outperform most of other sophisticated measures [16, 21, 27]. DTW provides elastic matching of two sequences while Euclidean distance is brittle since it only allows one-to-one point matching. It is widely accepted that DTW is state-of-the-art sequence measure. However, distance measures are not the entire story of a classification method. A classifier implicitly relies on both a distance measure and a classification strategy. Different classification strategies lead to different performance, even with the same distance measures. For example, the performance of 1-Nearest Neighbor (1-NN) could be different from that of K-Nearest Neighbor (K-NN) depending on the data domain.

In the plain input space, it is well-known that DTW is generally superior to Euclidean distance. However, *in the scenario of Kernel Machines, is it true that DTW outperforms Euclidean distance?* The answer is not straightforward as in the 1-NN classification, because kernel functions implicitly map input vectors to higher and even infinite dimensional feature space where the separability is greatly enhanced.

Kernel Machines, referred to as Support Vector Machines (SVMs), use the minimum structure risk as classification strategy and leave the distance measure flexible by the kernel tricks. The kernel tricks of SVM enable it to be a universal learning machine. For linear kernel, the implicit distance measure is the Euclidean distance. For the Gaussian RBF kernel, the distance is also Euclidean but in Gaussian form. The performance of SVMs with linear kernel and RBF kernel is usually superior to that of 1-NN even with the same Euclidean distance measure. However, *is it true that SVM with RRF kernel outperforms 1-NN classifier with DTW distance?*

Since SVM is state-of-the-art classifier, it is a natural extension to plug sequence measures into kernel machines for sequence classification. A so-called Gaussian DTW (GDTW) kernel was proposed for sequence classification with applications in online handwriting recognition and speech recognition [4, 24]. The kernel is defined as $k_\ell(x, y) = exp(-\frac{D_\ell(x,y)}{\sigma^2})$, where $D_\ell(\cdot, \cdot)$ denotes the DTW distance. Intuitively, the advantages of DTW can be utilized: elastic and robust matching of sequences, tolerance of different length which are common in sequential patterns. However, the time complexity for a single DTW calculation is $O(w * n)$ where n is the length of sequence and w is the width of band restriction. If DTW kernel is used in SVM classification, the DTW distances between input sequence and every support vector have to be computed in the SVM testing phase. Therefore, the time complexity is the first problem that DTW kernel has to face. Furthermore, the more important concern is, *is the DTW kernel a qualified SVM kernel? If not, why?* While counter-examples can be found to prove that DTW kernel is not Positive Definite Symmetric (PDS) numerically, little is known why DTW can not be PDS theoretically.

The study in the paper is to address the concerns above. The rest of this paper is organized as follows. In section 2, commonly used sequence measures *Lp* norms and DTW are briefly introduced. In section 3, kernel functions and DTW in kernel machines are firstly described. Then, we analyze DTW kernel and prove that it is not PDS theoretically. Extension on proof and related works will also be discussed. In section 4, experimental results are reported in comparing DTW and RBF in kernel classification. Finally, conclusion remarks are summarized in section 5.

2 Sequence Measures and Dynamic Time Warping

For sequence analysis, one of the core problems is how to define (dis)similarity measures. *Lp* norms and DTW are commonly used measures. Given two sequences $x = [x_1, x_2, \cdots, x_n]$ and $y = [y_1, y_2, \cdots, y_n]$, *Lp* norm is defined as $Lp(x, y) = (\sum_{i=1}^{n} \|x_i - y_i\|^p)^{\frac{1}{p}}$. It reduces to the commonly used Euclidean norm when $p = 2$. L_2 norm is optimal in the Maximum Likelihood sense when measurement errors are independent, identically distributed Gaussian. Thus, it has been widely applied in sequence matching and indexing [29]. *Lp* norms are fast to compute and easy for indexing. However, they assumes one-to-one point matching and thus are brittle in handling sequential patterns whose elements are non-linearly misaligned. Dynamic Time Warping has advantages over *Lp* norms in its elastic and robust matching. DTW is a method to find an optimal match between two given sequences (e.g. time series).

To compute the DTW distance $D_\iota(x, y)$ with $x = [x_1, x_2, \cdots, x_n]$ and $y = [y_1, y_2, \cdots, y_m]$, we can first construct an n-by-m matrix, as shown in Fig. 1. Then, we find a *path* in the matrix which starts from cell $(1, 1)$ to cell (n, m) so that the average cumulative cost along the path is minimized. If the path passes cell (i, j), then the cell (i, j) contributes $cost(x_i, y_j)$ to the cumulative cost. The *cost* function can be defined flexibly depending on the application, typically, $cost(x_i, y_j) = \|x_i - y_i\|^2$. This path can be determined using the dynamic programming, because the recursive equation holds: $D_\ell(i, j) = cost(x_i, y_j) + min\{D_\ell(i-1, j), D_\ell(i-1, j-1), D_\ell(i, j-1)\}$.

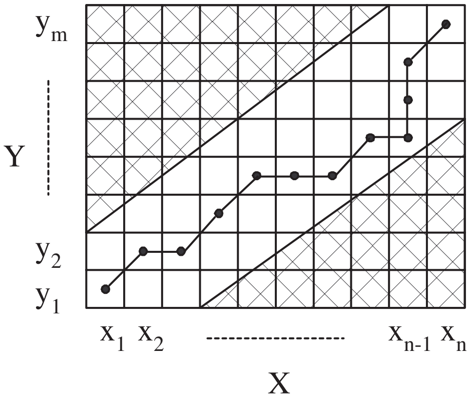

Figure 1. The warping path determined by DTW in the n-by-m matrix has the minimum cumulative cost. The marked area is the band restriction that path cannot go. The path indicates the optimal alignment: (x_1, y_1), (x_2, y_2), (x_3, y_2), $\cdots$, (x_{n-1}, y_{m-3}), (x_{n-1}, y_{m-2}), (x_{n-1}, y_{m-1}), (x_n, y_m).

The path may goes several cells horizontally along the x-axis or vertically along the y-axis, which makes the matching between the two sequences not strictly one-to-one but one-to-many. The warping path implicitly stretches both sequence x and y to same length and the cumulative cost is the Euclidean distance of the the stretched sequences. The stretching is indicated by the alignment path: if a point x_i in x corresponds to h points in y, then x_i is implicitly duplicated h times, because the cumulative cost is summed up by all the local costs of the points on the alignment path. The stretching is same with sequence y. Therefore, we can imagine there exists two "stretching" functions for a single DTW matching. Let ψ_x^y denotes the stretching function of x when matching with sequence y. And similarly, ψ_y^x. Note that both ψ_x^y and ψ_y^x map x and y to L_{xy}-dimensional Euclidean space. Note that the dimension L_{xy} also depends on x and y, since the warping path pertains to the matching. Thus, $D_\ell(x, y) = \|\psi_x^y(x) - \psi_y^x(y)\|^2$.

3 Kernels in Support Vector Machines

The training of SVMs is to solve a constrained quadratic optimization on the kernel matrix K which is constructed by computing the kernel function k between pairwise training samples, i.e, $K(i, j) = k(v_i, v_j)$, where v_i is the training vector $(i = 1, 2, \cdots, N)$. Let us first recall the definition of kernels.

Definition 3.1. Let X be a non-empty set. A function k :

$X \times X \rightarrow \Re$ is a kernel on X if there exists a Hilbert space H and a feature map $\phi : X \rightarrow H$ such that for all $x, y \in X$, $k(x, y) =< \phi(x), \phi(y) >$. H is called a feature space of k.

The kernels implicitly map the vectors in the input space to higher dimensional space where the dot products are computed using the kernel functions directly. Gaussian RBF kernel functions are among the most commonly used kernels, which is defined as:

$$k_{\sigma,d}(x, y) = exp(-\frac{\sum_{i=1}^{d} \|x_i - y_i\|^2}{\sigma^2}) \qquad (1)$$

Without loss of generality, we only consider real vector $x, y \in \Re^d$ here. Note that we use width σ and Euclidean dimension d in to denote the RBF kernel, because the kernel is determined by the two parameters.

It is well known that the Gaussian function is a kernel [2, 25]. Its mapping function ϕ maps n vectors $v_1, v_2, ..., v_n$ to n-dimensional space. $\phi(v_1)$, $\phi(v_2)$, ..., $\phi(v_n)$ are linearly independent and thus they span an n-dimensional subspace of Hilbert space. A Gaussian kernel is defined on a domain of infinite cardinality (the size of set X has no restriction). Therefore, it can map the vectors to a infinite dimensional space [20].

However, only PDS kernels are admissible to the standard SVMs since the Mercer condition must be satisfied to guarantee the optimal convergence of SVM training. With PDS kernels, the kernel matrix is convex and thus the training of SVM can reach the optimal solution. Gaussian kernels are typical PDS kernels [1] and was suggested to use in standard SVM [10].

Definition 3.2. Let X be a non-empty set. A kernel function $k : X \times X \rightarrow \Re$ is a Positive Definite Symmetric (PDS) kernel on X if it is symmetric and $\sum_{i,j=1}^{n} c_i c_j k(v_i, v_j) \geq 0$ for all $n \geq 0$, $\{v_1, v_2, ..., v_n\} \subseteq X$ and $\{c_1, c_2, ..., c_n\} \subseteq \Re$.

Definition 3.3. Let X be a non-empty set. A kernel function $k : X \times X \rightarrow \Re$ is a Negative Definite Symetric (NDS) kernel on X if it is symmetric and $\sum_{i,j=1}^{n} c_i c_j k(v_i, v_j) \leq 0$ for all $n \geq 0$, $\{v_1, v_2, ..., v_n\} \subseteq X$ and $\{c_1, c_2, ..., c_n\} \subseteq \Re$ with $\sum_{i=1}^{n} c_i = 0$.

A theorem clarifies the the relation between NDS and PDS and provides a way to construct PDS from NDS: *k is NDS $\Leftrightarrow$ $exp(-tk)$ is a PDS for all $t > 0$*[7]. Euclidean distance is NDS, that it is why Gaussian RBF is PDS [15].

By Linear algebra, k is a PDS kernel if and only if the kernel matrix K with $K(i, j) = k(v_i, v_j)$ is symmetric and all the eigenvalues of K are non-negative [6]. However, it is difficult to analyze the eigenvalues of an arbitrary matrix unless we numerically compute it [22]. Although counter-examples can be found to prove some kernels are not PDS, but only analytical method can discover why and how the PDS conditions are not satisfied.

3.1 DTW in kernel

Provided the kernel tricks of SVM and robustness of DTW, it is tempting to substitute DTW distance for the Euclidean distance in the RBF kernel. The concern is whether the defined kernel is a qualified kernel. It was claimed that the DTW kernel is a PDS kernel [23]. However, we found that the proof is incorrect. Let us review the proof, where the DTW kernel is defined as:

$$k_\ell(x, y) = exp(-\frac{D_\ell(x, y)}{\sigma^2}) \qquad (2)$$

where k_ℓ denotes the DTW kernel function, and D_ℓ denotes the DTW distance (we adjust some symbols for the sake of description, readers are referred to the original article for the details of the proof). Since D_ℓ is equivalent to the Euclidean distance of the stretched sequences, the authors stated that $k_\ell(x, y) = exp(-\frac{\|\psi_x(x)-\psi_y(y)\|^2}{\sigma^2})$, where ψ_x and ψ_y are the mapping functions implicitly used by DTW to stretch the two sequences with optimal alignment. The authors were aware of that the mapping depends on the particular sequence. However, they failed to recognize that the mapping depends on both sequences, that is why we denote the mapping function as ψ_x^y instead of ψ_x to indicate the mapping on x when x is matched against y. Moreover, they ignored that the RBF kernel is determined by both σ and the dimension d. When x matches to y, the two are stretched to length d. But when x matches to some other sequence instead of y, they are usually stretched to some length different from d. Therefore, the proof in [23] is invalid.

Even the proof was misleading, it is believed that the DTW kernel is not a PDS kernel since simple counterexamples can be found [11]. However, no analytical work has been done so far to prove that the DTW kernel is not a PDS kernel and little is known why the DTW kernel can not be PDS, to the best of our knowledge. One might intuitively think the reason lies in that DTW is not metric (no triangle relation). We will show that being a Hilbertian metric is sufficient but not necessary condition to construct a Gaussian kernel.

To complete a theoretical proof, we need to recall the definition and property of of Reproducing Kernel Hilbert Space (RKHS).

Definition 3.4. Let X be a non-empty set and H a Hilbert function space over X. H consists of functions which map X into $\Re$. The space H is called a RKHS over X if for $\forall x \in X$ the Dirac delta functional $\delta_x : H \rightarrow \Re$ is continuous, where $\delta_x(f) = f(x), f \in H$. A function $k : X \times X \rightarrow \Re$ is a reproducing kernel of H if we have $k(., x) \in H, \forall x \in X$ and the reproducing property $f(x) =< f, k(., x) >, \forall f \in H$.

Hilbert space is a generalized Euclidean space that is not restricted to finite dimensions. RKHS is a Hilbert function space, i.e., the elements in RHHS are functions, while regular Hilbert space consists of vectors. The reproducing kernels are kernels since $\phi : X \rightarrow H$ defined by $\phi(x) = k(.,x)$ is a feature map of k. Given a kernel, it is well known that the feature map and the feature space are not uniquely determined. However, according to the Moore-Aronszajn theorem [3], a PDS kernel uniquely determines a RKHS and vice versa. We will start from this point and prove that the DTW kernel is not a PDS kernel.

Theorem 3.1. *Let X be the space of real-value sequences. The function defined as (2) is not a PDS kernel function.*

Proof. Suppose x, y, z are three arbitrary sequences in X. We know that $k_\ell(x,y) = exp(-\frac{\|\psi^y_x(x)-\psi^x_y(y)\|^2}{\sigma^2}$. Since ψ^y_x and ψ^x_y stretch the two sequences to the same length d, we can consider $\psi^y_x(x), \psi^x_y(y) \in \Re^d$. Thus, $k_{\sigma,d}(\psi^y_x(x), \psi^x_y(y)) = k_\ell(x,y)$. For all $\sigma \in \Re$, $d \in N$, RBF kernel $k_{\sigma,d}$ is a reproducing kernel which uniquely determines a RKHS. Let us denote the RKHS determined by $k_{\sigma,d}$ as RKHS$_d$. Similarly, for $k_\ell(x,z)$, x and z are stretched to a d'-dimensional space. Thus, we have a corresponding RBF kernel $k_{\sigma,d'}$ which is the reproducing kernel of RKHS$_{d'}$. There exist two cases in the relation between RKHS$_d$ and RKHS$_{d'}$:

i)if $d \neq d'$, then RKHS$_d$ is different from RKHS$_{d'}$ because they are uniquely determined by $k_{\sigma,d}$ and $k_{\sigma,d'}$ respectively. On the otherside, if the k_ℓ is a PDS kernel,then RKHS$_d$=RKHS$_{d'}$. This leads to contradiction.

ii)if $d = d'$, then RKHS$_d$=RKHS$_{d'}$. If k_ℓ is PDS then there exist a unique RKHS over X reproduced by k_ℓ. Let denote RKHS$_d$ as H_ℓ. Recall that for any reproducing kernel k, the feature map between X and RKHS is $\phi(x) := k < \cdot, x >$ [25].

To match x and y using DTW, we have

$$\begin{aligned} k_\ell(x,y) &= k_{\sigma,d}(\psi^y_x(x), \psi^x_y(y) \\ &=< k_{\sigma,d} < \cdot, \psi^y_x(x) >, k_{\sigma,d} < \cdot, \psi^x_y(y) >> \end{aligned}$$

Thus, x is mapped to $k_\ell < \cdot, x >= k_{\sigma,d} < \cdot, \psi^y_x(x) >$ in H_ℓ. However, when matching x and z using DTW, we have

$$\begin{aligned} k_\ell(x,z) &= k_{\sigma,d}(\psi^z_x(x), \psi^x_z(z) \\ &=< k_{\sigma,d} < \cdot, \psi^z_x(x) >, k_{\sigma,d} < \cdot, \psi^x_z(z) >> \end{aligned}$$

Which means x is mapped to $k_\ell < \cdot, x >= k_{\sigma,d} < \cdot, \psi^z_x(x) >$. Therefore, $k_{\sigma,d} < \cdot, \psi^y_x(x) >$ must be equivalent to $k_{\sigma,d} < \cdot, \psi^z_x(x) >$. We substitute the latter to $k_\ell(x,y)$ and get:

$$\begin{aligned} k_\ell(x,y) &=< k_{\sigma,d} < \cdot, \psi^z_x(x) >, k_{\sigma,d} < \cdot, \psi^x_y(y) > \\ &= k_{\sigma,d}(\psi^z_x(x), \psi^x_y(y) \\ &= exp(-\frac{\|\psi^z_x(x) - \psi^x_y(y)\|^2}{\sigma^2}) \end{aligned}$$

This means the DTW distance between x and y is $\|\psi^z_x(x) - \psi^x_y(y)\|^2$, while real distance should be $\|\psi^y_x(x) - \psi^x_y(y)\|^2$. The two are equal only under conditions: i) $\psi^z_x = \psi^y_x$, or ii) there exist more than one warping paths in the cost matrix between x and y. The condition obviously does not hold in DTW matching, because x, y and z are arbitrarily chosen from X and DTW finds the *optimal* matching path. Therefore, $\|\psi^y_x(x) - \psi^x_y(y)\|^2 \neq \|\psi^z_x(x) - \psi^x_y(y)\|^2$ also leads to contradiction. □

From the proof, we can see that the stretching functions depend on the pair of sequences that are being matched, which is the main reason that a unique RKHS can not be found. On the other side, if some stretching functions map all the vectors to the same metric space, the kernel can be PDS. We can derive the following corollary.

Corollary 3.2. *Let X be a non-empty finite set and $\forall x \in X$, function f_x maps x to a d-dimensional metric space $\Re^d$ associated with Hilbertian metric distance measure D, then a PDS kernel function can be defined as:*

$$k_c(x, x') = exp(-\frac{D^2(f_x(x), f_{x'}(x'))}{\sigma^2}), \forall x, x' \in X \quad (3)$$

Here we use f_x and $f_{x'}$ to indicate that the mapping function for each instance in the X can be different from other functions, as long as they map the instances to the same metric space. We also generalize the Euclidean distance to any metric distance D. For any metric space (X, D), the space is also a Hilbert space. There is a theorem stating that *a metric space (X, D) embeds in a Hilbert space $\Leftrightarrow$ D^2 is Negative Definite Symmetric (NDS)* [2]. We also have the theorem that *D is NDS $\Leftrightarrow$ $exp(-\frac{D^2(\cdot,\cdot)}{\sigma^2})$ is a PDS* [7]. With the support of the two theorems, we can derive the following proof.

Proof. Let X^f be the metric set mapped from X by f_x, $\forall x \in X$. Define a PDS kernel as $k_D(e, e') = exp(-\frac{D^2(e,e)}{\sigma^2}), \forall e, e' \in X^f$. Then, k_D has a unique RKHS$_D$ and the feature mapping is $\phi_D(e) := k_D < \cdot, e >$, which means k_D is PDS over X^f. We can define a feature map for X as $\phi_c(x) = \phi_D(f_x(x) = \phi_D \cdot f_x(x)$. Therefore, the function space consists of $\phi_c(x)$ is a RKHS on X. So k_c is PDS over X. □

The corollary indicates that it is safe to plug any metric distance function to the Gaussian form to construct a PDS kernel (in this case, we can let the function f_x and $f_{x'}$ map vector to itself). f_x and $f_{x'}$ can be considered as a "preprocessing" functions. As long as the preprocessing functions independently map the original data to a metric space where metric function D is used as distance function, then kernel in (3) is PDS suitable for SVM. For instance, in SVM classification, we usually preprocess components of the feature vectors to some range, typically, [-1,1]. The preprocessing is implicitly a mapping function that maps the original data space to a new metric space. The corollary above guarantees that the preprocessing does not cause violation of the Mercer conditions because all the input feature vectors are preprocessed to the same metric space.

The corollary also implies that Hilbertian metrics are sufficient but not necessary to construction PDS kernels. Some distance measures which does not obey triangle relation can also construct a PDS kernel. For example, coefficient of determination R^2 can be used as a distance measure:

$$R^2(x, y) = \frac{[\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})]^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2} \qquad (4)$$

R^2 (the squared correlation Pearson's r) is the goodness-of-fit for linear regression [28]. It is known that correlation r and R^2 does not satisfy the triangle inequality. But a kernel defined as follows is PDS.

$$k_R(x, y) = exp(tR^2(x, y)), t > 0 \qquad (5)$$

Let $\nu(x)$ denotes the mean-deviation normalization, $\nu(x) = (x - \overline{x})/\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2}$. It is not difficult to show $R^2(x, y) = 2 - 2||\upsilon(x) - \upsilon(y)||^2$. Thus, $k_R(x, y) = exp(2t - 2t||\upsilon(x) - \upsilon(y)||^2) = exp(2t)exp(-2t||\upsilon(x) - \upsilon(y)||^2)$. Considering the υ as a preprocessing function, the k_R is PDS obviously according to the corollary.

3.2 Related Works

Like DTW, Edit distance is used in many application. It finds the minimal cost to transforming one string into another. It was first explicitly proved to be not NDS [7]. However, the proof is given basically by counter-examples instead of theoretical analysis. Extending the proof on DTW with trivial modification can conclude that the following function can not be PDS:

$$k_e(s, s') = exp(-tD_e(s, s')) \qquad (6)$$

where, D_e denotes Edit distance, s and s' are strings over alphabet Σ. Edit distance also implicitly stretch two strings to the same length and the stretching depend on the pair of strings. Once the a pair strings are stretched to the same vector space, the Hamming distance instead of Euclidean distance is used. The Hamming distance is also metric. Due to limitation of space, the proof is omitted here. From DTW and Edit distances, we can see that elastic matching based distances can not be converted to PDS. We suspect that elastic matching is not even a kernel, let alone PDS kernel. Of course, this is still any open problem.

Table 1. The datasets used on the experiments

Datasets	# of classes	# of training samples	# of testing samples	Length
Gun-Point	2	50	150	150
ECG	2	100	100	96
Lighting2	2	60	61	637
Yoga	2	300	3000	426

Table 2. The classification results of the four approaches

Datasets	RBF-SVM	DTW-SVM	DTW-NN	1-NN
Gun-Point	0.96 ($\sigma = 0.03, C = 50$)	0.94($\sigma = 3, C = 20$)	0.913	0.913
ECG	0.97 ($\sigma = 0.1, C = 50$)	0.83 ($\sigma = 0.2, C = 30$)	0.88	0.88
Lighting2	0.754 ($\sigma = 0.001, C = 30$)	0.737 ($\sigma = 0.01, C = 30$)	0.869	0.754
Yoga	0.854 ($\sigma = 0.013, C = 90$)	0.8 ($\sigma = 2, C = 20$)	0.854	0.83

4 Experiments

While DTW kernel is not PDS in general, the kernel matrix could be positive definite given a training dataset at hand (recall that the kernel is defined over non-empty set X which can be of any size). So one might argue we still can plug DTW distance to SVM for classification to take advantages of DTW's elastic matching. Our experiments compared the performance of DTW kernel and RBF kernel to see whether DTW still has advantages over Euclidean distance in kernel machines. Four datasets were used to conduct experiments. The four datasets, namely, Gun-Point, Lightning-2, Yoga and ECG have been extensively used in time series experiments [17]. Table 1 summarizes the description of the datasets. The toolbox adopted was the Matlab SVM routines[9] based on LIBSVM [5]. Programmes were written in Matlab to perform classification implementing the following approaches: 1) SVM with standard RBF kernel (RBF-SVM);2) SVM with DTW kernel (DTW-SVM); 3)Nearest neighbor classifier with DTW distance (DTW-NN); 4)1-nearest neighbor(1-NN). The kernel parameters, σ and C, were determined by cross validation.

The classification results are summarized in Table 2. Obviously, the RBF-SVM outperforms the DTW-SVM in all datasets. Interestingly, the DTW-SVM is even beat by the 1-NN except in the Gun-Point dataset. In general, the RBF-SVM achieves better performance than DTW- SVM , NN-DTW and 1-NN (except in the Lighting2, the DTW-NN has higher accuracy). Overall, the performance of the DTW-SVM is the worst. So it can be seen that the DTW kernel is not suited for SVM.

To further compare their generalization ability, we recorded the ratios of the support vectors with respect to the number of training samples in DTW-SVM and RBF-

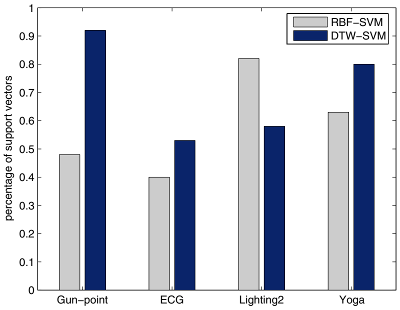

Figure 2. The percentages of support vectors in RBF-SVM training and DTW-SVM training on dataset Lighting2.

SVM. Figure 2 shows the percentage of the support vectors for both RBF-SVM and DTW-SVM on the four datasets. Lower percentage indicates lower generalization error, because the error is bounded by the ratio, though the bound is loose [26]. On three of the four datasets except the Lighting2 dataset, the DTW-SVM training generates more support vectors than the RBF-SVM, which means lower generalization ability. This is consistent with the classification results in Table 2.

5 Conclusions and future work

The DTW outperforms Euclidean distance in plain input space but not in kernel machines. We theoretically prove that the DTW kernel is not PDS and empirically show that DTW kernel does not lead to good performance due to poor generalization ability on a variety of datasets. Therefore, although some elastic matching measures are promising in plain input space, cautions must be taken when applying these measures in kernel machines.

Our future work will try to find a way to analyze the performance of DTW and other elastic measures in kernel machines rather than via empirical experiments. A possible theoretical explanation of the the poor performance remains in our future work. Also, it might be possible to force the DTW to comply with the Mercer's conditions by modifying the dynamic alignment calculation [8]. The String Kernels [18, 19] are inspiring examples that allow dynamic programming but still obey kernel conditions. We believe this research direction will be fruitful.

References

[1] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoér. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821C837, 1964.

[2] N. Aronszajn. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44:522–536, 1938.

[3] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[4] C. Bahlmann, B. Haasdonk, and H. Burkhardt. On-line handwriting recognition with support vector machines - a kernel approach. *The 8th International Workshop on Frontiers in Handwriting (IWFHR)*, pages 49–54, 2002.

[5] C.Chang and C. Lin. Libsvm: a library for Support Vector Machines. 2001.

[6] C. Cortes, P. Haffner, and M. Mohri. Positive definite rational kernels. *The 16th Annual Conference on Computational Learning Theory*, 2003.

[7] C. Cortes, P. Haffner, and M. Mohri. Rational kernels: Theory and algorithms. *The Journal of Machine Learning Research*, 5:1035–1062, 2004.

[8] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui. A kernel for time series based on global alignments. *32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP*, 2:413–416, 2007.

[9] S. Gunn. Support vector machine for classification and regression. *Technical Reprot, Image Speech and Intelligent Reseach Group, University of Southampton*, 1997.

[10] I. Guyon, B. Boser, and V. Vapnik. Automatic capacity tuning of very large VC-dimension classifiers. volume 5, pages 147–155, 1993.

[11] B. Haasdonk and D. Keysers. Tangent distance kernels for support vector machines. *The 16th International Conference on Pattern Recognition*, pages 864–868, 2002.

[12] Y. Huang and P. Yu. Adaptive query processing for time-series data. *The 5th International Conference on Knowledge Discovery and Data Mining*, pages 282–286, 1999.

[13] P. Indyk, N. Koudas, and S. Muthukrishnan. Identifying representative trends in massive time series data sets using sketches. *The 26th International Conference on Very Large Data Bases*, pages 363–372, 2000.

[14] K. Kalpakis, D. Gada, and V. Puttagunta. Distance measures for effective clustering of arima time-series. *The IEEE International Conference on Data Mining*, pages 273–280, 2001.

[15] Y. Katznelson. *An introduction to harmonic analysis*. John Wiley and Sons, New York, 1968.

[16] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[17] E. Keogh, X. Xi, L. Wei, and C. Ratanamahatana. The ucr time series classification/clustering. *http://www.cs.ucr.edu/ eamonn/time_series_data*.

[18] C. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for SVM protein classification. *Advances in Neural Information Processing Systems*, 2003.

[19] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 419-444:337–404, 2002.

[20] C. A. Micchelli. Algebraic aspects of interpolation. *Proceedings of Symposia in Applied Mathematics*, 36:81–102, 1986.

[21] S. Park, S. Kim, and W. Chu. Segment-based approach for subsequence searches in sequence databases. *The 16th ACM Symposium on Applied Computing*, pages 248–252, 2001.

[22] Y. Saad. *Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms*. Wiley, New York, 1992.

[23] H. Shimodaira, K. Noma, M. Nakai, and S. Sagayama. Support vector machine with dynamic time-alignment kernel for speech recognition. *The 7th European Conference on Speech Communication and Technology*, pages 1841–1844, 2001.

[24] H. Shimodaira, K. Noma, M. Nakai, and S. Sagayama. Dynamic time-alignment kernel in Support Vector Machine. *Advances in Neural Information Processing Systems*, pages 921–928, 2002.

[25] I. Steinwart, D. Hush, and C. Scovel. An explicit description of the reproducing kernel hilbert space of gaussian RBF kernels. *Technical report, Los Alamos National Laboratory*, 2004.

[26] V. Vapnik and O. Chapelle. Bounds on error expectation for support vector machines. *Neural Computation*, 12:2013–2036, 2000.

[27] C. Wang and X. Wang. Supporting content-based searches on time series via approximation. *The 12th International Conference on Scientific and Statistical Database Management*, pages 69–81, 2000.

[28] J. Wooldridge. *Introductory Econometrics: a modern approach*. South-Western College Publishing, 2nd edition, 1999.

[29] B. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary *Lp* norms. *The 26th International Conference on Very Large Databases*, pages 385–394, 2000.