Dynamic Time Warping and the (Windowed) Dog-Keeper Distance

Jörg P. Bachmann(✉) and Johann-Christoph Freytag

Humboldt-Universität zu Berlin, 10099 Berlin, Germany
joerg.bachmann@informatik.hu-berin.de,
freytag@dbis.informatik.hu-berlin.de

Abstract. Finding similar time series is an important task in multimedia retrieval, including motion gesture recognition, speech recognition, or classification of hand-written letters. These applications typically require the similarity (or distance) measure to be robust against outliers and time warps. Time warps occur if two time series follow the same path in space, but need specific time adjustments. A common distance measure respecting time warps is the dynamic time warping (DTW) function. The edit distance with real penalties (ERP) and the dog-keeper distance (DK) are variations of DTW satisfying the triangle inequality. In this paper we propose a novel extension of the DK distance called windowed dog-keeper distance (WDK). It operates on sliding windows, which makes it robust against outliers. It also satisfies the triangle inequality from the DK distance. We experimentally compare our measure to the existing ones and discuss the conditions under which it shows an optimal classification accuracy. Our evaluation also contributes a comparison of DK and DTW. For our experiments, we use well-known data sets such as the cylinder-bell-funnel data set and data sets from the UCI Machine Learning Repository.

Keywords: Time series · Metric · Multimedia retrieval

1 Introduction

Many applications require to find similar time series to a given pattern. One common application of finding similar time series is multimedia retrieval, including motion gesture recognition, speech recognition, and classification of handwritten letters. All these tasks have in common that the time series of same classes (e.g., same spoken words or same gestures) follow the same path in space, but have some temporal displacements. Another example is tracking the GPS coordinates of two cars driving the same route from A to B. Although we want these time series to be recognized as being similar, driving style, traffic lights, and traffic jams might result in large temporal differences. Distance functions such as dynamic time warping (DTW) [11], edit distance with real penalties (ERP) [4], and the dog-keeper distance (DK) [6] respect this semantic requirement.

C. Beecks et al. (Eds.): SISAP 2017, LNCS 10609, pp. 127–140, 2017.
DOI: 10.1007/978-3-319-68474-1_9

Another requirement for similarity functions is their computational performance since it is common to compare a sample time series to a large set of time series. To improve performance we might improve the computation time of one time series comparison or we might reduce the number of comparisons. Assuming that SETH [3] holds, Bringmann and Künnemann proved that there is no algorithm computing the exact value of DTW in less than quadratic time [3]. Similar results were proven for the DK distance [2] and the edit distance [1]. However, we are usually not interested in the exact distance values, but in the set of the nearest neighbours. A common approach for pruning elements as possible candidates are lower bounds to the distance function.

Keogh and Ratanamahatana exhaustively compared nine different time series distance functions including DTW and ERP on 38 time series data sets coming from different domains [5]. They also compared eight different time series representations including Discrete Fourier Transformation (DFT) and Symbolic Aggregate approXimation (SAX) [10]. In their work, they investigated contradictory claims about the effectiveness of different time series distance functions and representations. Their first major insight is that there is little difference in the effectiveness between different time series representations excluding some rare cases. They say there is no clear winner for the choice of the time series distance function, although elastic distance functions, such as DTW, ERP, LCSS, or EDR are more accurate, especially on small data sets.

To the best of our knowledge, DTW has not been compared to the DK distance. If DTW is the time warping equivalent to the L_1-norm, then the DK distance is the equivalent to the L_∞-norm and thereby more sensitive to noise or outliers within time series. On the other hand, we could observe a speed-up by an order of magnitude in our experimental evaluation. Why does the DK distance perform much better although the algorithm is quite similar to that of DTW? Can we improve the robustness of the DK distance?

The first contribution of our paper is the windowed DK distance (WDK), which is a modification of the DK distance to satisfy the triangle inequality. We evaluate the performance of the four time warping distance functions DTW, ERP, DK, and WDK by comparing the results of k-nearest neighbour classifiers on four different multimedia time series data sets coming from different domains. The second contribution is that we also investigate the reason for the low computation time of the DK and WDK distance functions.

The rest of this paper is structured as follows. Section 2 introduces basic terms and notations and reviews the time series distance functions DTW, ERP, and DK. Section 3 defines the WDK distance function and provides an algorithm for its computation. Section 4 evaluates these four distance functions on four multimedia time series data sets. Section 5 concludes the paper.

2 Preliminaries and Concepts

This section introduces basic notations and concepts used in this paper.

It is hard to find an open access proof for the triangle inequality of the DK distance in modern mathematical language. Therefore we provide a new proof in this section that shows this well known fact again. This also proves the triangle inequality for the WDK distance proposed in this paper.

Basic Notation: With $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geqslant c}$ we denote the set of non-negative integers, the set of reals, and the set of all reals $\geqslant c$, for some $c \in \mathbb{R}$, respectively. An $m \times n$ matrix is denoted by $A = (a_{i,j})$. Given a matrix A, $A_{i,j}$ denotes the element in the i-th row and j-th column.

By $\mathbb{R}^k$, for $k \in \mathbb{N}$, we denote the set of all vectors of length k. For a vector $v \in \mathbb{R}^k$ we write v_i for the entry at position i.

For mappings $f : A \longrightarrow B$ and $g : B \longrightarrow C$, we denote the image of f as $f(A) := \{f(x) \mid x \in A\}$ and $g \circ f : x \mapsto g(f(x))$ the concatenation of g and f. Furthermore, $\inf f$ and $\sup f$ are the infimum and the supremum of $f(A)$ respectively.

Norms and Metric Spaces: By $\|\cdot\|_p$, for $p \in \mathbb{R}_{\geqslant 1}$, we denote the well known L_p*-norm* on $\mathbb{R}^k$; i.e., $\|v\|_p = \left(\sum_{i=1}^{k} |v_i|^p\right)^{1/p}$ for all $v \in \mathbb{R}^k$.

Recall that a *pseudo metric space* $(\mathbb{M}, d)$ consists of a set $\mathbb{M}$ and a distance function $d : \mathbb{M} \times \mathbb{M} \longrightarrow \mathbb{R}_{\geqslant 0}$ satisfying the following axioms:

$$\begin{aligned} \forall\ x, y \in \mathbb{M}\ &: d(x,y) = d(y,x). \\ \forall\ x, y, z \in \mathbb{M}\ &: d(x,z) \leqslant d(x,y) + d(y,z). \end{aligned}$$

A *metric space* is a pseudo metric space which also satisfies $\forall\ x, y \in \mathbb{M}\ : d(x,y) = 0 \iff x = y$. Note that if $\|\cdot\|$ is an arbitrary vector norm and $d(\cdot,\cdot)$ is defined as $d(u,v) := \|u - v\|$, then $(\mathbb{R}^k, d)$ is a metric space. By d_p, for $p \in \mathbb{R}_{\geqslant 1}$, we denote the usual L_p-distance, i.e., the particular distance function with $\mathsf{d}_p(x,y) = \|x - y\|_p$.

Time Series: A time series T of length ℓ over a metric space $\mathbb{M}$ is a sequence $T = (t_1, \cdots, t_\ell)$ with $t_i \in \mathbb{M}$ for $1 \leqslant i \leqslant \ell$. We denote $\mathtt{Tail}(T) := (t_2, \cdots, t_n)$ as the time series when removing first element. In the rest of the paper, we consider $\mathbb{M} = \mathbb{R}^k$ for some $k \in \mathbb{N}$. We denote time series with the letters S, T, and R.

Time Series Distances: The algorithms for the computation of DTW, ERP, and DK are very similar. They differ in how they handle a time warping step and whether they take the maximum along a warping path or sum up these values. DTW and ERP sums the values up while the DK distance takes the maximum.

For a formal definition, let $S = (s_1, \cdots, s_m)$ and $T = (t_1, \cdots, t_n)$ be two time series, gap a globally constant element (0 as proposed by [4]), and $\mathsf{d}(s,t)$ a distance function for the elements of the time series. The well known distance function DTW is defined as follows.

$$\mathtt{DTW}(S, ()) = \infty \quad \mathtt{DTW}((), T) = \infty \quad \mathtt{DTW}((s), (t)) = \mathsf{d}(s,t)$$

$$\mathtt{DTW}(S,T) = \min \begin{cases} \mathsf{d}(s_1,t_1) + \mathtt{DTW}(\mathtt{Tail}(S), \mathtt{Tail}(T)) \\ \mathsf{d}(s_1,t_1) + \mathtt{DTW}(S, \mathtt{Tail}(T)) \\ \mathsf{d}(s_1,t_1) + \mathtt{DTW}(\mathtt{Tail}(S), T) \end{cases}$$

ERP differs from DTW by including gap elements to the time series on warping steps.

$$\texttt{ERP}(S,()) = \infty \quad \texttt{ERP}((),T) = \infty \quad \texttt{ERP}((s),(t)) = \texttt{d}(s,t)$$

$$\texttt{ERP}(S,T) = \min \begin{cases} \texttt{d}(s_1,t_1) + \texttt{DTW}(\texttt{Tail}(S),\texttt{Tail}(T)) \\ \texttt{d}(s_1,\texttt{gap}) + \texttt{DTW}(S,\texttt{Tail}(T)) \\ \texttt{d}(\texttt{gap},t_1) + \texttt{DTW}(\texttt{Tail}(S),T) \end{cases}$$

The DK distance is similar to DTW and differs by taking the maximum distance along a warping path instead of the sum.

$$\texttt{DK}(S,()) = \infty \quad \texttt{DK}((),T) = \infty \quad \texttt{DK}((s),(t)) = \texttt{d}(s,t)$$

$$\texttt{DK}(S,T) = \min \begin{cases} \max\{\texttt{d}(s_1,t_1), \texttt{DK}(\texttt{Tail}(S),\texttt{Tail}(T))\} \\ \max\{\texttt{d}(s_1,t_1), \texttt{DK}(S,\texttt{Tail}(T))\} \\ \max\{\texttt{d}(s_1,t_1), \texttt{DK}(\texttt{Tail}(S),T)\} \end{cases}$$

Note that ERP and DK satisfy the triangle inequality and therefore are metric distance functions [4,7]. See Fig. 1 for sketches of the behaviour of these distance functions.

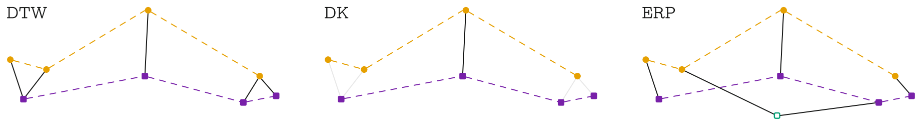

Fig. 1. Example time series with example warping paths sketching the behaviour of DTW (left), DK (center), and ERP (right). Distances between states are marked with solid lines while the circled and squared time series are connected using dashed lines. DTW sums up the distances along the warping path (all solid lines). DK is the largest distance along the warping path (longest solid line). ERP sums up the distances along the warping path (all solid lines). However, when warping (second circle from the left and third square from the right), states are compared to the gap element (empty square).

Algorithm 1 shows a pseudo code for computing the DK distance between two time series similar to the algorithm proposed by Eiter and Mannila [6]. We extended the algorithm considering a threshold as third parameter for early abandoning. The idea of early abandoning works the same in all algorithms for the mentioned time series distance functions. After computing the next row (or column) of the matrix D, the minimum value in that row is a lower bound for the final distance value. If that value already exceeds the threshold, the algorithm stops and returns the lower bound.

Dog-Keeper is a Metric: Satisfying the triangle inequality might be an opportunity for indexing the data using metric index structures. In the following we

Algorithm 1. Pseudo Code for the Dog-Keeper Distance with Early Abandoning

```
Input: $S = (s_1, \cdots, s_\ell)$, $T = (t_1, \cdots, t_m)$, $\tau$
Output: Lower bound for the dog-keeper distance

$D_{1,1} = \mathtt{d}(s_1, t_1)$
if $D_{1,1} \geqslant \tau$
   return $D_{1,1}$
for $i$ in $2, \cdots, \ell$
   $D_{i,1} = \max\{D_{i-1,1}, \mathtt{d}(s_i, t_1)\}$
for $j$ in $2, \cdots, m$
   $\varepsilon = \infty$
   for $i$ in $1, \cdots, \ell$
      $pred = \min\{D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\}$
      $D_{i,j} = \max\{\mathtt{d}(s_i, t_j), pred\}$
      $\varepsilon = \min\{\varepsilon, D_{i,j}\}$
   if $\varepsilon \geqslant \tau$
      return $\varepsilon$
return $D_{\ell,m}$
```

want to provide a new proof in modern mathematical language that shows that the dog-keeper distance satisfies the triangle inequality. Therefore, we prove the triangle inequality for th Fréchet distance. Since the dog-keeper distance is the discrete special case of the Fréchet distance, the proof also holds for the dog-keeper distance.

Let $\mathbb{M} := \mathbb{R}^k$ be the space of states, $\mathtt{d} : \mathbb{M} \times \mathbb{M} \longrightarrow \mathbb{R}_{\geqslant 0}$ be a metric on all states. We denote the set of all (piecewise continous) curves over $[0,1] \subset \mathbb{R}$ by

$$\mathcal{T} := \{f \colon [0,1] \longrightarrow \mathbb{M}\}$$

and the set of all time warps over [0,1] by

$$\Sigma := \{\sigma \colon [0,1] \longrightarrow [0,1]\},$$

where all $\tau \in \Sigma$ are continuous, strictly monotonically increasing, and $\inf \tau = 0$, $\sup \tau = 1$. For $f, g \in \mathcal{T}$, let $\delta_\infty(f,g) := \max_{x \in [0,1]} \mathtt{d}(f(x), g(x))$ be the maximum distance of f and g.

Definition 1 (Fréchet Distance). *Let $f, g \in \mathcal{T}$ be two curves over $[0,1]$. The Fréchet distance $\mathtt{DK}$ of f and g is defined as*

$$\mathtt{DK}(f,g) := \inf_{\sigma,\tau \in \Sigma} \delta_\infty(f \circ \sigma, g \circ \tau)$$

Using this notation we prove the following theorem.

Theorem 1. *The Fréchet distance $\mathtt{DK}$ satisfies the triangle inequality, i.e.,*

$$\forall f, g, h \in \mathcal{T} \colon \mathtt{DK}(f,h) \leqslant \mathtt{DK}(f,g) + \mathtt{DK}(g,h).$$

To prove the triangle inequality, we first prove the following lemma showing that the δ_∞ distance does not change when applying the same temporal adjustment to both curves. The second lemma then reduces the search to all warping functions applied to one time series only.

Lemma 1. *Let $f, g \in \mathcal{T}$ be two arbitrary curves and $\sigma \in \Sigma$ be an arbitrary time warp. Then, the following equation holds:*

$$\delta_\infty(f, g) = \delta_\infty(f \circ \sigma, g \circ \sigma)$$

Proof. Consider the mapping

$$\begin{aligned} \theta\colon [0,1] &\longrightarrow \mathbb{R}_{\geqslant 0} \\ x &\longmapsto \mathsf{d}(f(x), g(x)). \end{aligned}$$

Then,

$$\begin{aligned} \delta_\infty(f, g) &= \sup\left(\theta([0,1])\right), \text{and} \\ \delta_\infty(f \circ \sigma, g \circ \sigma) &= \sup\left(\theta \circ \sigma([0,1])\right) \end{aligned}$$

Since θ([0,1]) $= \theta(\sigma([0,1]))$, the desired equation $\delta_\infty(f, g) = \delta_\infty(f \circ \sigma, g \circ \sigma)$ follows. □

Lemma 2. *Let $f, g \in \mathcal{T}$ be two arbitrary curves. Then the following equation holds:*

$$\mathit{DK}(f, g) = \inf_{\sigma \in \Sigma} \delta_\infty(f, g \circ \sigma)$$

Proof. Consider two sequences $(\sigma_i)_{i \in \mathbb{N}}$ and $(\tau_i)_{i \in \mathbb{N}}$ with $\sigma_i, \tau_i \in \Sigma$ for $i \in \mathbb{N}$, such that

$$\delta_\infty(f \circ \sigma_i, g \circ \tau_i) \xrightarrow{i \to \infty} \mathtt{DK}(f, g).$$

Since each σ_i is invertable, Lemma 1 can be applied on $\delta_\infty(f \circ \sigma_i, g \circ \tau_i)$ with σ_i^{-1}, i.e. we obtain

$$\begin{aligned} \delta_\infty(f, g \circ \tau_i \circ \sigma_i^{-1}) &= \delta_\infty(f \circ \sigma_i \circ \sigma_i^{-1}, g \circ \tau_i \circ \sigma_i^{-1}) \\ &= \delta_\infty(f \circ \sigma_i, g \circ \tau_i) \xrightarrow{i \to \infty} \mathtt{DK}(f, g). \end{aligned}$$

Thus, we have a sequence $(\theta_i)_{i \in \mathbb{N}} := (\tau_i \circ \sigma_i^{-1})_{i \in \mathbb{N}}$ with $\theta_i \in \Sigma$ for $i \in \mathbb{N}$, such that $\delta_\infty(f, g \circ \theta_i) \xrightarrow{i \to \infty} \mathtt{DK}(f, g)$.

On the other hand,

$$\inf_{\sigma, \tau \in \Sigma} \delta_\infty(f \circ \sigma, g \circ \tau) \leqslant \inf_{\theta \in \Sigma} \delta_\infty(f, g \circ \theta).$$

Hence, $\mathtt{DK}(f, g) = \inf_{\theta \in \Sigma} \delta_\infty(f, g \circ \theta)$. □

Proof (*Proof of Theorem* 1). Consider some arbitrary but fixed $f, g, h \in \mathcal{T}$. Since $\mathtt{DK}(f, g) = \inf_{\sigma \in \Sigma} \delta(f, g \circ \sigma)$ (Lemma 2), an infinite sequence $(\sigma_i)_{i \in \mathbb{N}}$ exists with $\sigma_i \in \Sigma$ for all $i \in \mathbb{N}$, such that

$$\delta_\infty(f, g \circ \sigma_i) \xrightarrow{i \to \infty} \mathtt{DK}(f, g).$$

Analogously, a sequence $(\tau_i')_{i \in \mathbb{N}}$ with $\tau_i' \in \Sigma$ for all $i \in \mathbb{N}$ exists, such that

$$\delta_\infty(g, h \circ \tau_i') \xrightarrow{i \to \infty} \mathtt{DK}(g, h).$$

Considering the sequence $(\tau_i)_{i \in \mathbb{N}}$ with $\tau_i = \tau_i' \circ \sigma_i \in \Sigma$ and using Lemma 1, we obtain

$$\delta_\infty(g \circ \sigma_i, h \circ \tau_i) = \delta_\infty(g, h \circ \tau_i') \xrightarrow{i \to \infty} \mathtt{DK}(g, h).$$

Recall that $(\mathcal{T}, \delta_\infty)$ is a metric space, thus the triangle inequality holds for each $i \in \mathbb{N}$:

$$\delta_\infty(f, h \circ \tau_i) \leqslant \delta_\infty(f, g \circ \sigma_i) + \delta_\infty(g \circ \sigma_i, h \circ \tau_i)$$

Since $\mathtt{DK}(f, h) = \inf_{\tau \in \Sigma} \delta_\infty(f, h \circ \tau)$, we obtain the triangle inequality:

$$\mathtt{DK}(f, h) \leqslant \lim_{i \to \infty} \delta_\infty(f, h \circ \tau_i) \leqslant \mathtt{DK}(f, g) + \mathtt{DK}(g, h)$$ □

3 Windowed Dog-Keeper Distance

If there is one outlier in a time series, then this outlier dominates the DK distance, i.e. it dominates the maximum along a path through the matrix in Algorithm 1. Hence, the DK distance is not robust against outliers. In the case of DTW or ERP, the error of the outlier is relatively small compared to the sum of all small errors. One of our contributions is the windowed dog-keeper distance described below.

By comparing sliding windows with the L_1-norm instead of single elements, the same behaviour is possible for the DK distance. If there is an outlier within one time series, the error will not dominate the sum of distances within two sliding windows. For a formal definition, consider the sequence of sliding windows as a new time series.

Definition 2 (Windowed Time Series). *Let $n \in \mathbb{N}$ be an arbitrary window size and $T = (t_1, \ldots, t_\ell)$ be an arbitrary time series. The k-th n-window of T is the subsequence*

$$T_k^n = (t_k, \ldots, t_{k+n-1})$$

The n-windowed time series of T is the sequence

$$T^n = \left(T_1^n, \ldots, T_{k+n-1}^n\right)$$

Comparing two time series now is based on comparing windows. Here we might use the advantage of the L_1-metric to improve the robustness against outliers.

Definition 3 (Window Distance). *Consider two n-windows* $P = (p_1, \cdots, p_n)$ *and* $Q = (q_1, \cdots, q_n)$. *Then*

$$d(P, Q) = \sum_{i=1}^{n} d(p_i, q_i)$$

We now define the windowed dog-keeper distance (WDK).

Definition 4 (Windowed Dog-Keeper Distance). *Let S and T be two time series and n be an arbitrary window size. The n-windowed dog-keeper distance (n-WDK) of S and T is the dog-keeper distance of their n-windowed time series, i.e.*

$$\mathit{WDK}_n(S, T) := \mathit{DK}(S^n, T^n)$$

If it is clear from the context, we omit the parameter n.

Corollary 1. *The windowed distance is a metric.*

Note that the 1-WDK distance is equivalent to the DK distance, thus n-WDK can be seen a generalization of the DK distance.

The WDK distance is more robust against outliers as the experiments show. However, it comes with a price. The distance measure is less robust against local time warping, since the time series can drift apart within one window. Hence, the window size is a tuning parameter to choose between robustness against outliers and robustness against time warps. The larger the window size is, the more we gain robustness against outliers. With shrinking window size we increase the robustness against strong time warps.

Computation: When computing the WDK distance the naive way, there is a lot of redundancy. For example, computing the 2-window distances $d(S^1, T^1) = \mathtt{d}(s_1, t_1) + \mathtt{d}(s_2, t_2)$ and $d(S^2, T^2) = \mathtt{d}(s_2, t_2) + \mathtt{d}(s_3, t_3)$ each includes computing $\mathtt{d}(s_2, t_2)$. The first improvement of the WDK algorithm caches these values.

The second improvement optimizes the computation of the sum of the distance values along an n-window by first computing integral matrices. For time series S and T of length m and n respectively, the integral matrix $\int(S, T)$ is defined as

$$\int(S, T)_{i,j} = \begin{cases} \sum_{k=0}^{i-1} \mathtt{d}\left(s_{i-k}, t_{j-k}\right) & \text{if } i \leqslant j \\ \sum_{k=0}^{j-1} \mathtt{d}\left(s_{i-k}, t_{j-k}\right) & \text{else} \end{cases} \tag{1}$$

where $\int(S, T)_{i,j}$ is the entry in the i-th row and the j-th column. Less formally, we sum up the values of the matrix $(\mathtt{d}(s_i, t_j))$ along diagonals. The n-window distance $\mathtt{d}(T_i^n, S_j^n)$ is computed as a difference of two matrices:

$$\mathtt{d}(T_{i-n+1}^n, S_{j-n+1}^n) = \begin{cases} \int(S,T)_{i,j} - \int(S,T)_{i-n,j-n} & \text{if } i,j \geqslant n \\ \int(S,T)_{i,j} & \text{else} \end{cases} \tag{2}$$

Finally, the Fréchet distance is computed based on the window distance. Algorithm 2 represents the algorithm in pseudo code. Line 5 to 11 compute the integral matrix, line 14 to 16 compute the window distances. For time series of length ℓ and m, these sections have complexity $\mathcal{O}(\ell \cdot n)$. The rest of the code computes the Fréchet Distance similarly to Algorithm 1 but on the window distances thus the overall complexity is $\mathcal{O}(\ell \cdot n)$. Furthermore, the complexity does not depend on the window size.

Example 1. Consider the following example: $S = (1,2,1,5,6)$, $T = (2,1,6,5,6)$. When computing the 3-WDK distance function, the matrices in Algorithm will contain the following elements if they did not stop because of early abandoning:

$$I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 & 4 & 5 \\ 0 & 0 & 2 & 4 & 8 & 8 \\ 0 & 1 & 0 & 7 & 8 & 13 \\ 0 & 3 & 5 & 1 & 7 & 9 \\ 0 & 4 & 8 & 5 & 2 & 7 \end{pmatrix} \quad W = \begin{pmatrix} 7 & 8 & 13 \\ 1 & 6 & 9 \\ 5 & 2 & 5 \end{pmatrix} \quad D = \begin{pmatrix} 7 & 8 & 13 \\ 7 & 7 & 9 \\ 7 & 7 & 7 \end{pmatrix}$$

Thus, the 3-WDK distance of S and T is 7.

4 Experimental Evaluation

We evaluate the performance of the time series distance functions DTW, ERP, DK, and WDK on four data sets. Our first choice is the well-known cylinder-bell-funnel data set (CBF) as an example of noisy data. The other three data sets come from the UCI Machine Learning Repository [9]. We chose the following labeled multidimensional multimedia data sets: the Character Trajectories sata set (CT), the Spoken Arabic Digit data set (SAD), and the Australian Sign Language signs (High Quality) data set (ASL) [8].

Data Preparation: We prepared the data sets by normalizing them individually. The CT data set consists of three-dimensional time series holding the derivative of the trajectory and the pressure of the pen. We first integrated the derivative to retrieve the actual pen coordinates. The resulting time series have been normalized using the L_2-norm.

The Spoken Arabic Digits data set has been normalized using the L_1-norm. Furthermore, we removed the 23 shortest time series to assure that each time series has a length of at least 20 elements, such that we can evaluate the WDK distance for window sizes up to 20.

The ASL data set consists of 22-dimensional time series, 11 dimensions for each hand holding position, rotation and five finger bend information. We normalized the position information of the hands using the L_2 norm.

Algorithm 2. Pseudo Code for the n-Windowed Dog-Keeper Distance

```
Input: $S = (s_1, \cdots, s_\ell)$, $T = (t_1, \cdots, t_m)$, $\tau$
Output: Lower bound for the dog-keeper distance

// compute integral matrix $I_{i,j} = \int(S,T)_{i,j}$ as in Equation (1)
for $i$ in $0, \cdots, \ell$
    $I_{i,0} = 0$
for $j$ in $1, \cdots, m$
    $I_{0,j} = 0$
for $i$ in $1, \cdots, \ell$
    for $j$ in $1, \cdots, m$
        $I_{i,j} = I_{i-1,j-1} + \mathtt{d}(s_i, t_j)$

// compute the $n-$window distances $W_{i,j} = d(T_i^n, S_j^n)$ as in Equation (2)
for $i$ in $n+1, \cdots, \ell+1$
    for $j$ in $n+1, \cdots, m+1$
        $W_{i-n,j-n} = I_{i-1,j-1} - I_{i-n-1,j-n-1}$

// compute the DK distance as in Algorithm 1.
$D_{1,1} = W_{1,1}$
if $D_{1,1} \geqslant \tau$
    return $D_{1,1}$
for $i$ in $2, \cdots, \ell-n+1$
    $D_{i,1} = \max\{D_{i-1,1}, W_{i,1}\}$
for $j$ in $2, \cdots, m-n+1$
    $\varepsilon = \infty$
    for $i$ in $1, \cdots, \ell-n+1$
        $pred = \min\{D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}\}$
        $D_{i,j} = \max\{W_{i,i}, pred\}$
        $\varepsilon = \min\{\varepsilon, D_{i,j}\}$
    if $\varepsilon \geqslant \tau$
        return $\varepsilon$
return $D_{\ell-n+1,m-n+1}$
```

Retrieval Correctness: We use the data sets to evaluate the quality of the distance functions experimentally. Since we have chosen labeled time series, we can evaluate the correctness using a k-nearest neighbour classifier. We specifically ran a Leave-One-Out cross-validation on each data set. In order to evaluate the discriminability of the distance functions we ran the tests for different k from 1 to values larger than the class size.

Figure 2 shows that DTW has almost best retrieval results on the noisy CBF data set. ERP decreases in quality with increasing k. For small k, all distance

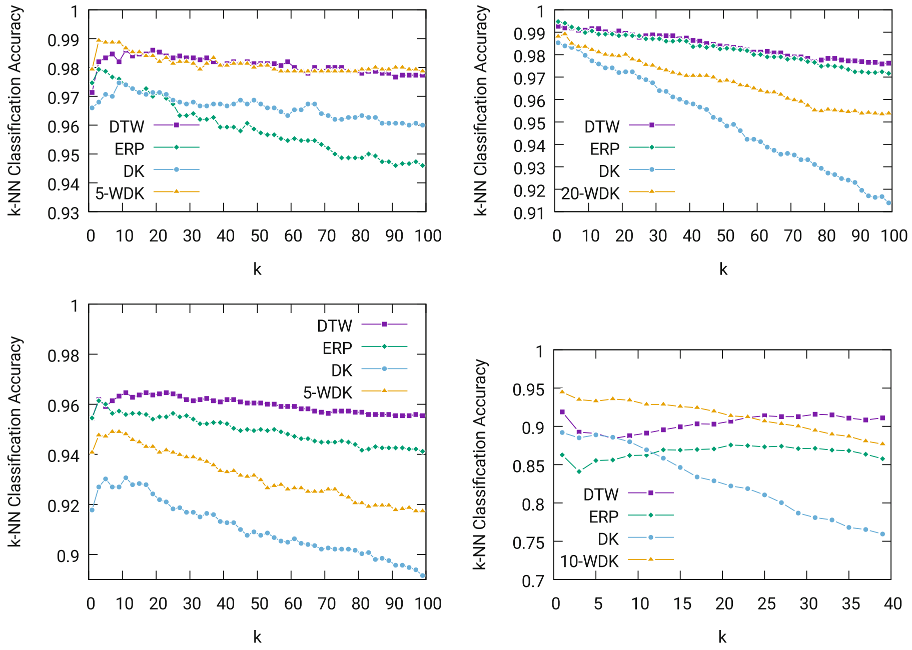

Fig. 2. Classification Accuracy on the CBF (top left), CT (top right), SAD (bottom left) and ASL (bottom right) data sets.

functions provide similar quality. We did not expect the WDK distance to perform well on that data set since it is very noisy (cf. Fig. 3).

In contrast to the CBF data set, Fig. 2 shows that the retrieval results decrease linearly with increasing k on the CT data set. Although there is nearly no difference in retrieval quality for small k, there is a clear tendency for large k. DTW and ERP have identical behaviour, while the DK is way behind. This experiment also shows that WDK improves the DK distance. Figure 3 shows two representational examples from the data set. We could not find any outliers in the data set and there is little need for warping. On the other hand, the distance between two points along the characters differ on long parts of the path, thus there are windows with a large distance to each other. This could be the reason for the good performance of DTW and ERP but the bad performance of DK and WDK.

Figure 2 shows the results for the SAD data set. The results are similar to those on the CT data set. The WDK distance improves the DK distance but loses against DTW and ERP.

Most interesting results for the WDK distance can be found in the ASL data set, shown in Fig. 2. Since both distance functions DTW and ERP drop to around 90% correctness, 10-WDK is nearly 95%. Another interesting observation here is that both "sum natured" functions DTW and ERP are increasing in correctness

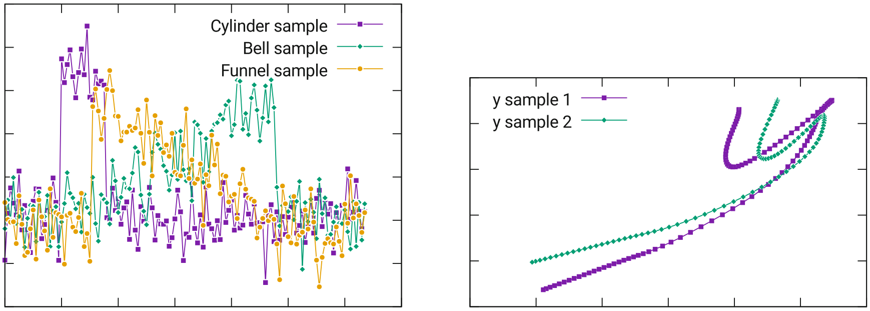

Fig. 3. Examples from the cylinder-bell-funnel (left) and the Character Trajectories (right) data sets.

with increasing k while both "max natured" functions DK and WDK decrease. We have not found a reasonable explanation yet, thus it remains future work.

Parameter Tuning: A disadvantage of the WDK distance is that it has a parameter (the window size) as we need to calibrate it for each data set. However, in all but the CBF data set, taking a window size of 25% of the mean time series length provided best results. For certain applications, the best parameter could be evaluated on a sample of the data set beforehand.

Figure 4 shows that the window size adjusts a trade-off as we expected. There is an optimal value and the classification Accuracy decreases monotone with diverging window size.

Computation Time: Table 1 shows the relative computation times with DTW as the base line for the 1-nearest neighbour classifier. Although the algorithms of all distance functions are quite similar the DK and WDK distance functions ran faster

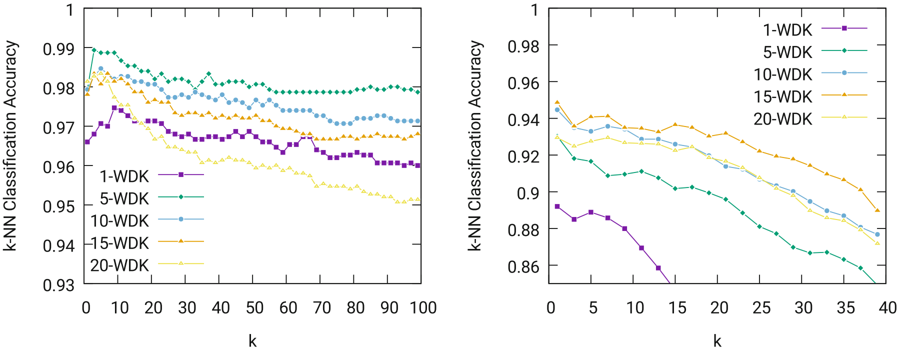

Fig. 4. Classification Accuracy on the CBF (left) and the Australian Sign Language (right) data sets for WDK with different window sizes.

by more than an order of magnitude. These differences can not be explained by implementation details. The only plausible explanation is the early abandoning.

Since DTW and ERP sum up the errors along the warping path, the probability for later abandoning increases. On the other hand, the DK distance takes the maximum value along the best warping path and therefore aborts computation most likely during the first step. We call the number of columns we need to compute before the computation can be aborted the point of early abandoning. The only exception is a value of 0 which means that the first elements of the time series are compared only.

Table 1. Computation time in relation to the computation time of DTW

	DTW	ERP	DK	WDK
CBF	1	0.89	0.23	0.13
Spoken digits	1	1.2	0.06	0.04
Signs	1	1.24	0.05	0.08
Character	1	1.39	0.13	0.06

Table 2 shows measurements of the number of comparisons which are aborted immediately after comparing the first elements of the time series on the ASL data set. It shows that 94.9% and 99.6% of the computations of the DK and WDK distance abort immediately, resp. The mean point of early abandoning for DTW and ERP is more than 10, which means that in most cases more than 10 columns of the matrix are filled.

Table 2. Point of early abandoning.

	DTW	ERP	DK	WDK
Immediate	0%	0%	94.9%	99.6%
Mean	10.8	13.3	0.39	0.21

5 Conclusion and Future Work

In this paper we compared the performance of different time warping distance functions on multimedia time series data sets. We have chosen data sets for motion gesture recognition, speech recognition, and classification of handwritten letters. This work extends existing evaluations by comparing the dog-keeper distance against DTW and ERP on these data sets. Although DTW has the best classification results on most data sets, we could show that the dog-keeper distance has nearly same results. For 1-nearest neighbour classification, the error rate of the dog-keeper distance was no more than 3% worse.

We also observed a significant difference in computation time. Our investigation showed that the reason is the very early abandoning.

We also improved the dog-keeper distance by comparing sliding windows instead of single elements. Our experimental evaluation shows that this modification did increase the classification correctness of the dog-keeper distance. On the Australian Sign Language data set (ASL), it even outperforms the other distance functions in retrieval quality. Furthermore, it inherited the property of early abandoning from the dog-keeper distance and even improved these values. On the Australian Sign Language data set, 99.6% of the comparisons already stopped after comparing the first elements of the time series. Hence, it seems there is no need for any further optimization using lower bounds.

It remains future work to investigate and compare to these functions with the Sakoe Chiba band [11] applied. We expect nearly the same behaviour from the dog-keeper and windowed dog-keeper distances. However, there are lower bounds to DTW with a Sakoe Chiba band applied which drastically improve retrieval times.

References

1. Backurs, A., Indyk, P.: Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). CoRR, abs/1412.0348 (2014)
2. Bringmann, K.: Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless SETH fails. CoRR, abs/1404.1448 (2014)
3. Bringmann, K., Künnemann, M.: Quadratic conditional lower bounds for string problems and dynamic time warping. CoRR, abs/1502.01063 (2015)
4. Chen, L., Ng, R.: On the marriage of Lp-norms and edit distance. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, vol. 30, pp. 792–803. VLDB Endowment (2004)
5. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endow. **1**(2), 1542–1552 (2008)
6. Eiter, T., Mannila, H.: Computing discrete Fréchet distance. Technical report, Technische Universität Wien (1994)
7. René Fréchet, M.: Sur quelques points du calcul fonctionnel. 22. Rendiconti del Circolo Mathematico di Palermo (1906)
8. Kadous, M.W.: Temporal classification: extending the classification paradigm to multivariate time series. Ph.D. thesis, School of Computer Science and Engineering, University of New South Wales (2002)
9. Lichman, M.: UCI machine learning repository (2013)
10. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Discov. **15**(2), 107–144 (2007)
11. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. In: Waibel, A., Lee, K. (eds.) Readings in Speech Recognition, pp. 159–165. Morgan Kaufmann Publishers Inc., San Francisco (1990)