A WASSERSTEIN SUBSEQUENCE KERNEL FOR TIME SERIES

Christian Bock[†] Matteo Togninalli[†] Elisabetta Ghisu
Thomas Gumbsch Bastian Rieck
Karsten Borgwardt

Machine Learning and Computational Biology Lab, D-BSSE, ETH Zurich, Switzerland

SIB Swiss Institute of Bionformatics, Switzerland

firstname.lastname@bsse.ethz.ch

[†]These authors contributed equally

The popular $\mathcal{R}$-convolution kernel framework is a powerful learning approach for structured data. However, when assessing the similarity of two time series through their subsequences, a simple but common instance of this framework can be meaningless. We utilize the power of optimal transport theory to propose a meaningful indefinite kernel for time series analysis that captures both local and global characteristics. We highlight the utility of our method by comparing it to state-of-the-art time series classification methods on a wide variety of data sets.[1]

1 INTRODUCTION

Applications of optimal transport (OT) theory in machine learning have recently soared and benefit a large variety of learning problems such as sorting [11], graph compression [18], or graph classification [33]. The general applicability of OT is underlined by its use in different learning approaches such as deep learning [34] or kernel-based learning [6]. One popular approach of the latter for data of discrete nature is the $\mathcal{R}$-convolution kernel framework [20] which identifies similarities of objects by the means of the similarities of their substructures. Substructures, or more precisely, subsequences, are the core concept of the majority of state-of-the-art time series classification (TSC) algorithms.

Time series are ubiquitous in numerous domains, ranging from astrophysics [30] to biomedical applications [5], with time series classification remaining a highly active research topic. TSC methods make use of extremely diverse methodologies, relying on the extraction of short, predictive subsequences [40], for example, or on distance measures such as dynamic time warping (DTW). We approach this topic from a different point of view, taking inspiration from the field of kernel methods. While some attempts were made to develop relevant kernel functions for TSC, successes in this area are limited. We hypothesise that this might be due to the fact that kernel function construction for time series suffers from two fundamental pitfalls. First, many similarity measures are either hypersensitive or insensitive to time shifts. Second, some time series subsequence kernel functions, representing simple instances of the $\mathcal{R}$-convolution framework [20], can be meaningless, as we will show in Section 2.

[1]Please note that this is an extended and modified version of a paper accepted at the 19th IEEE International Conference on Data Mining (ICDM).

In this paper, we introduce the Wasserstein Time Series Kernel (WTK), a kernel for time series that captures the similarity between subsequence distributions in addition to their pairwise similarities. WTK relies on notions from OT, a field increasingly popular in the machine learning community that provides similarity measures between probability distributions [28]. In the remainder of the paper, we describe the following contributions in detail: 1. We show that a straightforward application of simple instances of $\mathcal{R}$-convolution kernels to time series data can be meaningless. 2. We develop the first subsequence-based distance measure for time series that relies on the Wasserstein distance 3. We demonstrate its competitive classification performance in comparison to state-of-the-art methods.

2 Background

2.1 Kernel theory

Let $\mathcal{X}$ be a set with n elements and $\mathrm{k}\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a function that is symmetric and positive definite[2], i.e. $\sum_{i,j=1}^{n} c_i c_j \, \mathrm{k}(x_i, x_j) \geq 0$ for every $c_i \in \mathbb{R}$ and $x_i, x_j \in \mathcal{X}$. Then there exists a *Hilbert space* $\mathcal{H}$, i.e. a complete inner product space, and a mapping $\phi\colon \mathcal{X} \to \mathcal{H}$ such that $\mathrm{k}(\cdot, \cdot)$ can be equivalently expressed as $\mathrm{k}(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product in $\mathcal{H}$. The space $\mathcal{H}$ is also referred to as a *Reproducing Kernel Hilbert Space* (RKHS) because its inner product *reproduces* k. Since proving that positive definiteness holds for a kernel function can be challenging [35], alternative approaches that are not based on an RKHS were developed [27]. They are known as *Reproducing Kernel Kreĭn Spaces* (RKKS). In an RKKS, the positive definiteness requirement for the kernel function is dropped, so that kernels are allowed to be indefinite. Previous research [19] also showed that support vector machine (SVM) classifiers can use indefinite kernel matrices while maintaining favourable predictive performance.

2.2 Meaningless subsequence kernels

However, kernel construction can suffer from certain pitfalls, particularly when using simple instances of the $\mathcal{R}$-convolution kernel framework [20], which evaluates a base kernel between all substructures and aggregates them. Letting T, T' refer to two time series, $\mathcal{S}, \mathcal{S}'$ to their respective sets of subsequences, and k_{base} be a base kernel function, such a kernel takes the form of

$$\mathrm{k}\big(T, T'\big) := \frac{1}{|T| \cdot |T'|} \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}'} \mathrm{k}_{\text{base}}\big(s, s'\big). \tag{1}$$

Choosing k_{base} as a linear kernel will lead to

$$\begin{aligned} \mathrm{k}\big(T, T'\big) &= \frac{1}{|T| \cdot |T'|} \sum_{s \in \mathcal{S}} \sum_{s' \in \mathcal{S}'} s^\top s' \\ &\approx \frac{1}{|T| \cdot |T'|} \left(\sum_{s \in \mathcal{S}} s^\top \right) \left(\sum_{s' \in \mathcal{S}'} s' \right) \approx \overline{T}^\top \overline{T}', \end{aligned} \tag{2}$$

where the last approximation follows from the fact that in the respective sums over all subsequence feature vectors, almost all of the observations of length w, except for the *leading* $w - 1$ as well as the *trailing* $w - 1$ observations, will appear at all dimensions of the vectors in the sum. Hence, for many values of w, this $\mathcal{R}$-convolution kernel degenerates to a simple comparison of the means of two time series T and T'. The consequence of Eq. 2 is that, in particular for z-normalised data sets, which are the suggested default in time series analysis [29], the kernel becomes de facto *meaningless*. Our argumentation runs along the lines of the famous result [21] about the inherent meaninglessness of clustering time series subsequences. In

[2]For reasons of notational simplicity, we will not make a distinction between 'positive definite' and 'positive semi-definite' in this paper.

practice, when looking at some data sets from the 'UCR Time Series Classification Archive' [7], we observe that the values obtained from a $\mathcal{R}$-convolution linear kernel are close to zero for short subsequence lengths. As our experiments in Section 5 demonstrate, even increasing the subsequence length w does not result in competitive predictive performance.

3 Related work

3.1 Time series classification

There is a plethora of time series classification approaches, so we refer the reader to Bagnall et al. [2] for a comprehensive overview of methods. The de facto standard database for the benchmarking of time series classification algorithms is the 'UCR Time Series Archive' [7], a repository of 85 labelled time series data sets that was recently increased to 128 time series [13]. In addition to the methods assessed by Bagnall et al. [2], Wang et al. [38] established a baseline of neural network approaches, comprising a fully convolutional network as well as a residual network architecture, among others.

3.2 Kernel methods for time series

In light of the definiteness property discussion in Section 2.1, we discuss some definite and indefinite kernels for time series classification. The first kernel-based classification approaches comprise standard SVM kernels (linear, RBF) between whole time series [32]. For periodic patterns, several cross-correlation kernels are available [37]. Furthermore, some methods are based on DTW kernels [23] or general alignments of time series [10, 8], the former being indefinite in general. This lack of definiteness prompted an investigation into the impact of indefinite kernels on classification performance and lead to recursive edit distance kernel for TSC [26].

Closest to our current method is `KEMD` [12], which uses the Earth Mover's Distance [31] on histograms of the time series data points and evaluates it for EEG classification. While this kernel also partially relies on optimal transport, it fundamentally differs from ours, which uses subsequences rather than histograms (as we specify in Section 4.2). Finally, Cuturi et al. [10, 8] define an alignment kernel via the polytope of all possible alignments. Here, two subsequences are considered to be similar if they share a wide set of efficient alignments.

4 Our method

4.1 Optimal transport

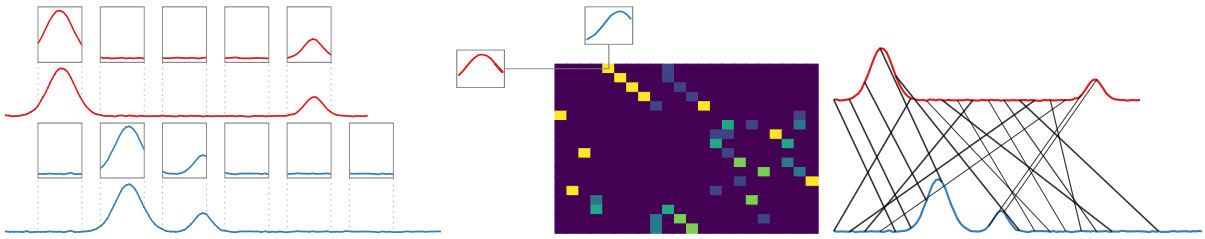
One of optimal transport's commonly-used methods is the *Wasserstein distance*. Given probability distributions on some metric space, the Wasserstein distance defines a metric between them. More precisely, let σ and μ be two probability distributions defined on a metric space $\mathcal{M}$ with some metric $\mathrm{dist}(\cdot, \cdot)$, which we refer to as the *ground distance*[3].

Definition 1. *Given $p \in \mathbb{R}_{>0}$, the p^{th} Wasserstein distance is defined as*

$$\mathrm{W}_p(\sigma, \mu) := \left(\inf_{\gamma \in \Gamma(\sigma, \mu)} \int_{\mathcal{M} \times \mathcal{M}} \mathrm{dist}(x, y)^p \mathrm{d}\gamma(x, y) \right)^{\frac{1}{p}}, \tag{3}$$

where $\Gamma(\sigma, \mu)$ is the set of all transportation plans $\gamma \in \Gamma(\sigma, \mu)$ over $\mathcal{M} \times \mathcal{M}$ with marginals σ and μ on the first and second factors, respectively.

[3] This terminology is used on purpose to distinguish it from the metric *induced* by the Wasserstein distance.



(a) Subsequence extraction (b) Transport plan (c) Transport plan visualisation

Figure 1: To measure the dissimilarity between two time series, our method proceeds in several steps. a First, all subsequences of the two time series are extracted based on a sliding window approach (here, not all subsequences are shown because their windows overlap). b After calculating the pairwise distance matrix between all subsequences, the optimal transport plan is computed. Its visualisation makes correspondences more readily visible. Yellow indicates a large fraction of transported mass (high subsequence similarity), whereas blue indicates a small fraction (low subsequence similarity). c Each line indicates a match between two subsequences, with the line being anchored at the respective beginning of the corresponding subsequences. The thickness of the line indicates the transport value. For clarity, only the largest transport values are shown.

In practice, we use a modified definition that can be described in terms of a matrix optimisation problem (see Definition 2). The Wasserstein distance satisfies the axioms of a metric, as long as $\mathrm{dist}(\cdot,\cdot)$ is a metric (see the monograph of Villani [36], Chapter 6, for a proof). Throughout the paper, we will focus on the 1^{st} Wasserstein distance, i.e. $p = 1$, and refer to it as *the* Wasserstein distance, unless noted otherwise.

The Wasserstein distance is intricately related to optimal transport problems [36], where the general goal is to find the most 'inexpensive' (in terms of predefined costs) way to transport the probability mass of one probability distribution σ to another probability distribution μ. It is possible to reformulate the Wasserstein distance as an optimisation problem between two matrices [31], making it more applicable to our setting.

Definition 2. *Let $X \in \mathbb{R}^{n\times m}$ and $Y \in \mathbb{R}^{n'\times m}$ be two matrices. We consider X and Y to represent sets of feature vectors of dimension m, but of varying cardinalities n and n'. The 1^{st} Wasserstein distance between X and Y is defined as*

$$\mathrm{W}_1(X,Y) := \min_{P\in\Gamma(X,Y)} \langle D, P\rangle_{\mathrm{F}}, \tag{4}$$

where D is an $n \times n'$ matrix containing the pairwise distances $\mathrm{dist}(x,y)$ for $(x,y) \in X \times Y$, P is the transport matrix, and $\langle\cdot,\cdot\rangle_{\mathrm{F}}$ is the Frobenius inner product.

The transport matrix P contains the fractions indicating how to transport the values from X to Y with the lowest effort. If we assume that the total mass to be transported is 1 and is evenly distributed across the elements of X and Y, then the values for the rows and columns of P must sum up to $1/n$ and $1/n'$, respectively.

4.2 A SUBSEQUENCE-BASED WASSERSTEIN KERNEL

We now define our novel subsequence-based Wasserstein kernel. Let $w \in \mathbb{N}_{>0}$ refer to a window width, or, equivalently, a subsequence length. Given a set of n time series $\mathcal{T} := \{T_1,\ldots,T_n\}$ we denote their set of length-w subsequences as $\mathcal{S} := \{\mathcal{S}_1,\ldots,\mathcal{S}_n\}$. For time series with a uniform length of m, the set $\mathcal{S}_i$ therefore contains $m - w + 1$ subsequences.

Definition 3 (Wasserstein time series kernel)**.** *Let T_i and T_j be two time series, and $s_{i1},\ldots,s_{iU}$ as well as $s_{j1},\ldots,s_{jV}$ be their respective subsequences. Moreover, let D be a $U\times V$ matrix that contains the pairwise*

distances of all subsequences, such that $D_{uv} := \mathrm{dist}(s_{iu}, s_{jv})$*, where* $\mathrm{dist}(\cdot,\cdot)$ *denotes the usual Euclidean distance. Following Definition 2, we have to solve the optimisation problem*

$$\mathrm{W}_1(T_i, T_j) := \min_{P \in \Gamma(T_i, T_j)} \langle D, P \rangle_{\mathrm{F}}, \tag{5}$$

which yields the optimal transport cost to transform T_i *into* T_j *by means of their subsequences. Then, given* $\lambda \in \mathbb{R}_{>0}$*, we define*

$$\mathrm{WTK}(T_i, T_j) := \exp(-\lambda \, \mathrm{W}_1(T_i, T_j)), \tag{6}$$

which we refer to as our Wasserstein-based subsequence kernel*; we will discuss its theoretical properties in Section 4.4.*

Since we consider a time series T_i to be represented by its set of subsequences $\mathcal{S}_i$, we will also write

$$\mathrm{W}_1(\mathcal{S}_i, \mathcal{S}_j) := \mathrm{W}_1(T_i, T_j) \tag{7}$$

and

$$\mathrm{WTK}(\mathcal{S}_i, \mathcal{S}_j) := \mathrm{WTK}(T_i, T_j) \tag{8}$$

to simplify the notation. During the calculation of WTK, the expression in Eq. 5, i.e. $\mathrm{W}_1(T_i, T_j)$, is a metric. It would therefore also be possible to use it in a k-nearest neighbour (k-NN) classifier. While we initially performed experiments to analyse this, we observed that k-NN classifiers yield substantially worse performance than the kernel defined according to Eq. 6.

In fact, we consider our use of the Wasserstein distance to be more similar to that of an $\mathcal{R}$-convolution kernel. Our argument is motivated by the observation that we can see Eq. 5 as a decomposition of the time series in terms of their subsequences. Furthermore, since $\mathrm{W}_1(\cdot,\cdot)$ is permutation-invariant, the order in which these subsequences are detected does *not* matter, so that Eq. 6 leads to an $\mathcal{R}$-convolution kernel with a single decomposition. This explains our preference for the kernel-based approach, as well as the favourable performance differences in a classification setting.

4.3 Intuition

Similar to shapelet-based methods, WTK makes use of the descriptive power of the subsequences of a time series. Figure 1 depicts the individual steps of our method, i.e. length-w subsequence extraction (Figure 1a) followed by the calculation of the optimal transport plan (Figure 1b).

The transport plan P that results from solving the optimisation problem in Eq. 5 can be seen as a map that assigns each length-w subsequence of the first time series (columns) to *at least* one length-w subsequence of the other time series (rows). Figure 1c shows how the obtained transport plan values map on the example time series. The formulation of the optimisation problem already accounts for different cardinalities in the respective sets. Our method is therefore applicable to time series of varying lengths, as depicted in the previous figure. In case the time series have the same length, this mapping is a bijection.

The obtained Wasserstein distance value is capable of better capturing the difference between the time series in terms of subsequence distributions as compared to merely summing the values of the transport plan. In the example, we can observe that the optimisation procedure selects the lowest distances between subsequences and aligns the respective peaks of the time series correctly.

4.4 Theoretical properties

Before using the similarity measure defined in Eq. 6 in a classical support vector machine, we have to prove that it satisfies the properties of a kernel. More precisely, if we want to have a kernel that belongs to an RKHS, we have to prove that it is positive definite. According to Feragen et al. [16, Theorem 5], this is equivalent to stating that for any data set, the symmetric matrix $\mathcal{D}$ whose entry (i, j) is of the form $\mathcal{D}_{ij} =$

$\mathrm{W}_1(T_i, T_j)$ is (conditionally) negative definite, which implies that it has at most *one* positive eigenvalue [3, Lemma 4.1.4, p. 163]. Feragen et al. [16, Theorem 4] showed that this implies that the metric space induced by Eq. 5 can be isometrically embedded into a Hilbert space.

Our empirical results indicate that for a few data sets and some configurations, we observe *more* than one positive eigenvalue in $\mathcal{D}$; the kernel matrix $\mathcal{K}$, whose entries are defined as $\mathcal{K}_{ij} = \mathrm{WTK}(T_i, T_j) := \exp(-\lambda\, \mathrm{W}_1(T_i, T_j))$, is therefore not positive definite. This leads us to conjecture that properties of the time series influence the induced metric, leaving us with several options:

(a) We can *enforce* the eigenvalue condition by calculating $\mathcal{L} := \mathcal{K} \cdot \mathcal{K}^\top$, where $\mathcal{K}$ refers to the $n \times n$ matrix with entries according to Eq. 6. Letting $y := \mathcal{K}^\top x$ for $x \in \mathbb{R}^n$, we then have $x^\top \mathcal{K}\mathcal{K}^\top x = x^\top \mathcal{K} y = y^\top y = \sum_{i=1}^n y_i \geq 0$, so $\mathcal{L}$ is positive definite. This is also known as the *empirical kernel*. It is computationally the easiest, requiring only an additional matrix multiplication. However, it changes the values between individual time series, and we observed in our experiments that the predictive performance suffers when compared with other options.

(b) We can *simplify* the matrix by subtracting all negative eigenvalues, leading to $\mathcal{L} := \mathcal{K} - \sum_i \lambda_i v_i v_i^\top$, where i ranges over the indices of the negative eigenvalues and v_i denotes their corresponding unit eigenvectors. By construction, this will set negative eigenvalues to zero, leaving us with a positive definite matrix. This is computationally harder, requiring a full eigendecomposition.

(c) We can *generalise* the Wasserstein distance to a 'softmin' of all possible transportation plans, which ensures that we obtain a positive definite kernel. However, it scales exponentially with the number of subsequences and is infeasible for all practical purposes [39].

(d) We can *sidestep* the eigenvalue condition by using algorithms that are capable of handling these *indefinite* matrices [27].

The majority of all data sets in our experiments resulted in a positive definite kernel matrix $\mathcal{K}$, making the options outlined above unnecessary. Nevertheless, to ensure classifier convergence, we employ a Kreĭn SVM [22] (following Option d), which is capable of handling positive definite and indefinite matrices. Unlike the other options, this one does not have to modify the kernel values at testing time. Hence, we refer to WTK as a *kernel*, with the added caveat that for some data sets, *the kernel matrix is indefinite*. We also tried Options (a) and (b) but none of them exhibited a significantly better performance.

COMPARING TO KEMD As mentioned in Section 3, despite some shared theoretical background, our method differs substantially from the Kernel Earth Mover's Distance (KEMD) method proposed by Daliri [12]: while KEMD can be considered a histogram intersection kernel [24] that treats each time series as a one-dimensional distribution of scalar values, our approach measures the distance between high-dimensional distributions of *subsequences*. We argue that it is therefore better suited to capture long-distance similarities of subsequences and time series.

4.5 EXTENSIONS

Our kernel easily permits changing the ground distance. Since previous work has shown that the choice of distance is crucial for obtaining suitable predictive performance [25], this could potentially further improve our experimental results. Another extension would be to calculate $\mathrm{WTK}(\cdot, \cdot)$ with subsequences *up to* a certain length w. While possible, this would require an innovative way to calculate distances between subsequences of different lengths. The commonly-used sliding Euclidean distance, which is typically employed for shapelet mining [40], is *not* a metric, as it does not satisfy the 'identity of indiscernibles'.

4.6 COMPLEXITY

The complexity of our method comprises the following parts: (a) Subsequence extraction, (b) Subsequence distance calculation, and (c) Wasserstein metric calculation. Letting n refer to the number of time series, we have at most $s := m - w + 1$ subsequences per time series. The extraction process is therefore dominated

by m, the length of the time series, leading to a total complexity of $\mathcal{O}(nm)$. This is a pre-processing step that we share with other methods, such as shapelet extraction methods [40].

Then, the following operations are performed for each pair of time series. Computing the distances between subsequences of two time series requires s^2 distance calculations, each of which has to process a sequence of length w. In the worst case, this calculation has a complexity of $\mathcal{O}\big(s^2 w\big)$, although it is possible to reduce this quite significantly, at least in the case of Euclidean distances, by re-using calculations.

Finally, evaluating Eq. 5 for two time series has a complexity of $\mathcal{O}\big(s^3 \log s\big)$ for an $s \times s$ input matrix [1]. Asymptotically, the runtime of all parts can be summarised as $\mathcal{O}\big(n^2 m^3 \log m\big)$, because m is an upper bound on the number of subsequences of a fixed length. This is a worst-case approximation and there are ways to obtain *near linear-time* approximate solutions for the Wasserstein distance, such as the approaches by Benamou et al. [4] or Cuturi [9], mostly involving Sinkhorn iterations [1]. In practice, for extremely long time series with a large number of subsequences, the use of the Sinkhorn approximation or one of its variants [1] could be beneficial. In our experimental setup, while a speed improvement could be observed, the accuracies obtained via a straightforward Sinkhorn approximation were not competitive with the results of the exact distance computation and hence were not included in our experiments.

5 Experiments

We perform all experiments on 85 data sets from the 'UCR Time Series Archive' [13]. Each data set consists of a predefined train/test split and the time series length differs across the data sets[4]. Our implementation uses `Python 3.7` and `POT`, the *Python Optimal Transport* library [17]. We make our code publicly available[5].

Training and evaluation We evaluate the classification accuracy on the test set and select the parameters on the training set via 5-fold cross validation using a Kreĭn SVM classifier [22] with the following parameter grid: $\gamma = \{10^{-5}, 10^{-4}, \ldots, 10^{3}\}$ (for the RBF kernel), $\lambda = \{10^{-4}, 10^{-3}, \ldots, 10\}$ (for WTK), $C = \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ (for the SVM classifier). We also vary the length w of the subsequences by checking values of w as $10\,\%$, $30\,\%$, and $50\,\%$ of the original time series length.

5.1 Comparison to other kernels

We compare our method with a standard linear and an RBF kernel based on subsequences in Figure 2. As outlined in Section 2.2, we expect the linear kernel to perform badly. By contrast, the RBF kernel has already shown favourable performance in previous research [32], but to our knowledge, we are the first to include it in a large-scale comparison. We would expect this kernel to perform better because it is capable of capturing non-linear patterns. However, the RBF kernel compares each pair of subsequences *independently*, whereas WTK is able to capture similarities between the *entire distributions* of subsequences of two time series.

We observe that our kernel outperforms the simple linear kernel in *all* cases, thereby giving a practical demonstration of the theoretical issues outlined in the introduction: a straightforward application of simple instances of the $\mathcal{R}$-convolution framework for subsequences of time series is meaningless. As for the RBF kernel, we outperform it on all but twelve data sets. The accuracy difference for the data sets in which the RBF kernel is better is negligible, and the average difference in predictive performance is only $\approx 2.2\,\%$. This demonstrates that the performance of our method is not caused by considering subsequences per se, but by considering the distribution of subsequence similarities.

[4] Please refer to `http://www.timeseriesclassification.com` for additional details.

[5] Please find a snapshot of our repository under `https://osf.io/dva3m/?view_only=6e01c6eceb23414690ced45dc5daa776` We plan on providing a GitHub repository as well, and in the meantime, we provide an anonymised version for the review process, which provides example code for reproducing a subset of our experiments.

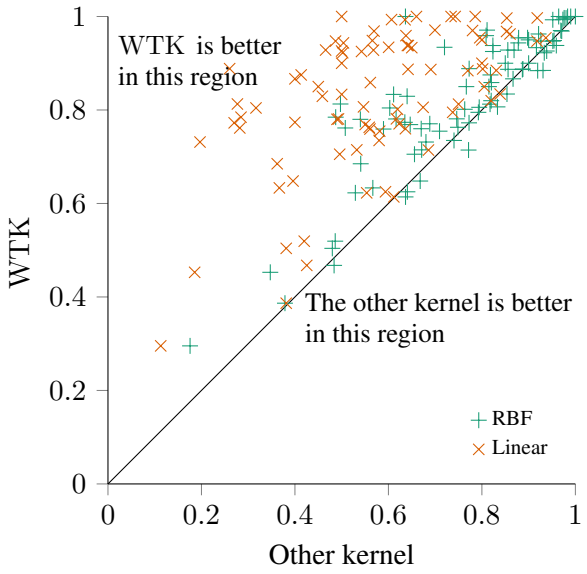


Figure 2: Comparison of the predictive accuracy of our method with the Linear and the RBF kernels for the 'UCR Time Series Archive' data sets.

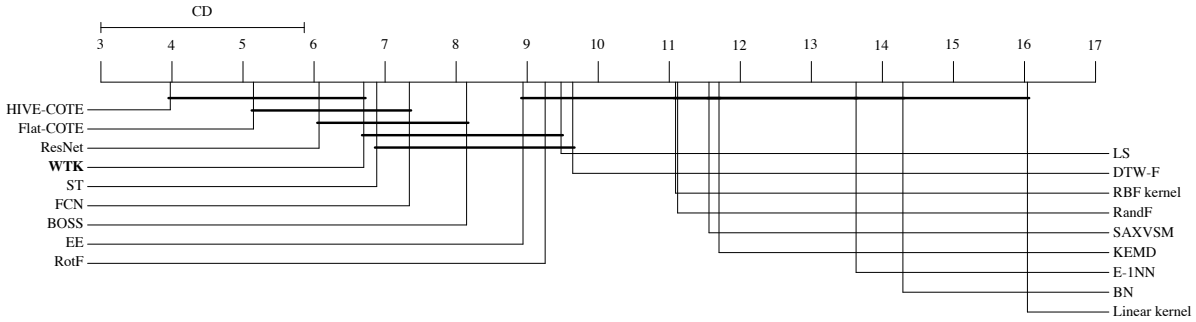


Figure 3: Critical difference plot, comparing our method (shown in bold) with several other methods. The scale indicates the average rank of each method in terms of test accuracy for all data sets. The classification performances of methods sharing horizontal bars are not significantly different. We observe that there is no statistically significant difference between the performance of our method and state-of-the-art ensemble methods.

5.2 Comparison to the state of the art

Next, we compare our method with 40 state-of-the-art methods in time series classification. To this end, we collected the accuracies of *all* published methods of the 'UCR Time Series Classification Repository' [7], as well as two neural network baselines [38] with their classification performances from [15]. For each data set, we picked the best-performing method on the published test set, referring to it as the respective state-of-the-art method. In total, we therefore compare our method with the best of 40 other methods. Our method outperforms *all* SOTA methods on six data sets. Overall, we exhibit at least *equal* accuracy as any state-of-the-art method on twelve data sets and are on a par with all SOTA methods on six data sets.

Statistical analysis To make the claims about the utility of our method more substantial, Figure 3 shows a *critical difference plot* [14] that depicts our method and several other methods. For a significance level of $\alpha = 0.05$, the plot shows that there is no statistically significant difference in the performance of our method and these best-performing classifiers [2]. The best-performing classifiers are either deep neural

networks or large ensembles and therefore heavily parametrised, hence showing a good promise for the generalisation performance of our method.

6 Conclusion

We developed a novel subsequence-based kernel that uses the Wasserstein distance as an effective similarity measure for time series classification. To prove the benefits of our method, we performed a large-scale evaluation on the 'UCR Time Series Archive' data sets that showed that our method outperforms some of the state-of-the-art time series classification algorithms while also displaying favourable generalisation properties. Currently, our method only considers fixed-size subsequence lengths and we plan to include varying-length subsequences in future work. However, there is a computational burden in their selection, as well as additional constraints because no commonly-used metric for comparing subsequences of varying lengths exists. Furthermore, we plan to explore how to extend our method to multivariate time series. Finally, effective subsequence preselection techniques will reduce the computational burden while potentially increasing the predictive performance. In conclusion, this work demonstrates the merits of OT-based kernels for time series analysis.

References

[1] J. Altschuler, J. Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30, pages 1964–1974. Curran Associates, Inc., 2017.

[2] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

[3] R. B. Bapat and T. E. S. Raghavan. *Nonnegative matrices and applications*. Cambridge University Press, Cambridge, UK, 1997.

[4] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[5] C. Bock, T. Gumbsch, M. Moor, B. Rieck, D. Roqueiro, and K. Borgwardt. Association mapping in biomedical time series via statistically significant shapelet mining. *Bioinformatics*, 34(13):i438–i446, 2018.

[6] M. Carrière, M. Cuturi, and S. Oudot. Sliced Wasserstein kernel for persistence diagrams. In *ICML*, pages 664–673, 2017.

[7] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The UCR time series classification archive, 2015. URL `http://www.cs.ucr.edu/~eamonn/time_series_data`.

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

[9] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NIPS)*, volume 26, pages 2292–2300. Curran Associates, Inc., 2013.

[10] M. Cuturi, J.-P. Vert, Ø. Birkenes, and T. Matsui. A kernel for time series based on global alignments. In *ICASSP*, volume 2, pages 413–416, 2007.

[11] M. Cuturi, O. Teboul, and J.-P. Vert. Differentiable sorting using optimal transport: The Sinkhorn cdf and quantile operator. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[12] M. R. Daliri. Kernel earth mover's distance for EEG classification. *Clinical EEG and Neuroscience*, 44(3):182–187, 2013.

[13] H. A. Dau, A. J. Bagnall, K. Kamgar, C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. J. Keogh. The UCR time series archive. *arXiv e-prints*, abs/1810.07758, 2018.

[14] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[15] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, pages 1–47, 2019.

[16] A. Feragen, F. Lauze, and S. Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *IEEE CVPR*, pages 3032–3042, 2015.

[17] R. Flamary and N. Courty. POT python optimal transport library, 2017. URL `https://github.com/rflamary/POT`.

[18] V. K. Garg and T. Jaakkola. Solving graph compression via optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[19] B. Haasdonk. Feature space interpretation of SVMs with indefinite kernels. *IEEE TPAMI*, 27(4): 482–492, 2005.

[20] D. Haussler. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz, 1999.

[21] E. Keogh and J. Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 8(2):154–177, 2005.

[22] G. Loosli, S. Canu, and C. S. Ong. Learning SVM in Kreĭn spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(6):1204–1216, 2015.

[23] A. Lorincz, L. Attila Jeni, Z. Szabo, J. F. Cohn, and T. Kanade. Emotional expression classification using time-series kernels. In *IEEE CVPR Workshops*, pages 889–895, 2013.

[24] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *IEEE CVPR*, pages 1–8, 2008.

[25] A. Mallasto, J. Frellsen, W. Boomsma, and A. Feragen. (q,p)-Wasserstein GANs: Comparing ground metrics for Wasserstein GANs. *arXiv e-prints*, art. arXiv:1902.03642, 2019.

[26] P.-F. Marteau and S. Gibet. On recursive edit distance kernels with application to time series classification. *IEEE Transactions on Neural Networks and Learning Systems*, 26(6):1121–1133, 2015.

[27] D. Oglic and T. Gärtner. Learning in reproducing kernel Kreĭn spaces. In *ICML*, pages 3859–3867, 2018.

[28] G. Peyré, M. Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[29] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*, pages 262–270, 2012.

[30] U. Rebbapragada et al. Finding anomalous periodic time series. *Machine learning*, 74(3):281–313, 2009.

[31] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[32] S. Rüping. SVM kernels for time series analysis. Technical Report 43, Technical University of Dortmund, 2001.

[33] M. Togninalli, E. Ghisu, F. Llinares-López, B. Rieck, and K. Borgwardt. Wasserstein Weisfeiler–Lehman graph kernels. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[34] I. O. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf. Wasserstein auto-encoders. In *6th International Conference on Learning Representations (ICLR)*, 2018.

[35] J.-P. Vert. The optimal assignment kernel is not positive definite. *arXiv e-prints*, art. arXiv:0801.4061, 2008.

[36] C. Villani. *Optimal transport: Old and new*, volume 338 of *Comprehensive Studies in Mathematics*. Springer, Heidelberg, Germany, 2008.

[37] G. Wachman, R. Khardon, P. Protopapas, and C. R. Alcock. Kernels for periodic time series arising in astronomy. In *Machine Learning and Knowledge Discovery in Databases*, pages 489–505, Heidelberg, Germany, 2009. Springer.

[38] Z. Wang, W. Ya, and T. Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1578–1585, 2017.

[39] L. Wu, I. En-Hsu Yen, F. Xu, P. Ravikumar, and M. Witbrock. D2KE: From distance to kernel and embedding. *arXiv e-prints*, art. arXiv:1802.04956, 2018.

[40] L. Ye and E. Keogh. Time series shapelets: A new primitive for data mining. In *KDD*, pages 947–956, New York, NY, USA, 2009. ACM.