frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 22 December 2021
doi: 10.3389/fams.2021.787455

Kernel Distance Measures for Time Series, Random Fields and Other Structured Data

Srinjoy Das[1]*, *Hrushikesh N. Mhaskar*[2] *and Alexander Cloninger*[3]

[1]School of Mathematical and Data Sciences, West Virginia University, Morgantown, WV, United States, [2]Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA, United States, [3]Department of Mathematics and Hağiğolou Data Science Institute, University of California, San Diego, San Diego, CA, United States

This paper introduces kdiff, a novel kernel-based measure for estimating distances between instances of time series, random fields and other forms of structured data. This measure is based on the idea of matching distributions that only overlap over a portion of their region of support. Our proposed measure is inspired by MPdist which has been previously proposed for such datasets and is constructed using Euclidean metrics, whereas kdiff is constructed using non-linear kernel distances. Also, kdiff accounts for both self and cross similarities across the instances and is defined using a lower quantile of the distance distribution. Comparing the cross similarity to self similarity allows for measures of similarity that are more robust to noise and partial occlusions of the relevant signals. Our proposed measure kdiff is a more general form of the well known kernel-based Maximum Mean Discrepancy distance estimated over the embeddings. Some theoretical results are provided for separability conditions using kdiff as a distance measure for clustering and classification problems where the embedding distributions can be modeled as two component mixtures. Applications are demonstrated for clustering of synthetic and real-life time series and image data, and the performance of kdiff is compared to competing distance measures for clustering.

Keywords: Kernels, time series, statistical distances, random fields, Motif detection

OPEN ACCESS

Edited by:
Yiming Ying,
University at Albany, United States

Reviewed by:
Xin Guo,
Hong Kong Polytechnic University,
Hong Kong SAR, China
Yunwen Lei,
Southern University of Science and
Technology, China

****Correspondence:***
Srinjoy Das
srinjoy.das@mail.wvu.edu

Specialty section:
This article was submitted to
Mathematics of Computation and Data
Science,
a section of the journal
Frontiers in Applied Mathematics and
Statistics

Received: *30 September 2021*
Accepted: *16 November 2021*
Published: *22 December 2021*

Citation:
Das S, Mhaskar HN and Cloninger A
(2021) Kernel Distance Measures for
Time Series, Random Fields and Other
Structured Data.
Front. Appl. Math. Stat. 7:787455.
doi: 10.3389/fams.2021.787455

1 INTRODUCTION AND MOTIVATION

Clustering and classification tasks in applications such as time series and image processing are critically dependent on the distance measure used to identify similarities in the available data. Distance measures are named as such because they may not satisfy all assumptions of a metric (e.g., they may induce equivalence classes of time series that are distance zero from one another, they may not satisfy the triangle inequality). In such contexts, several distance measures have been proposed in the literature:

- Point-to-point matching e.g. Euclidean distance or Dynamic Time Warping distance [1, 2]
- Matching features of the series e.g. autocorrelation coefficients [3], Pearson correlation coefficients [4], periodograms [5], extreme value behavior [6]
- Number of matching subsequences in the series [7]
- Similarity of embedding distributions of the series [8]

In this paper we consider distance measures for applications involving clustering, classification and related data mining tasks in time series, random fields and other forms of possibly non i. i.d data.

In particular, we focus on problems where membership in a specific class is characterized by instances matching only over a portion of their region of support. In addition, the regions where such feature matching occurs may not be overlapping in time, or on the underlying grid of the random field. Distance measures must take these data characteristics into consideration when determining similarity in such applications. Previously MPdist has been proposed as a distance measure for such time series datasets [7] which match only over part of their region of support and is constructed using Euclidean metrics. Inspired by MPdist, we propose a new kernel-based distance measure kdiff for estimating distances between instances of such univariate and multivariate time series, random field and other types of structured data.

For constructing kdiff, we first create sliding window based embeddings over the given time series or random fields. We then estimate a distance distribution by using a kernel-based distance measure between such embeddings over given pairs of data instances. Finally the distance measure used in clustering, classification and related tasks is defined by a pre-specified lower quantile of this distance distribution. This kernel-based distance measure based on such embeddings can also be constructed using the Reproducing Kernel Hilbert Space (RKHS) based Maximum Mean Discrepancy (MMD) previously discussed in [9]. Our kernel based measure kdiff can be considered as a more general distance compared to RKHS MMD for applications where class instances match only over a part of the region of support. More details about the connections between kdiff and RKHS MMD are provided later in the paper. We also note that the kernel construction in kdiff allows for data-dependent kernel construction similar to MMD [10–12], though we focus on isotropic localized kernels in this work and compare to standard MMD.

The rest of the paper is organized as follows. **Section 2** outlines the main idea and motivation behind the construction of our distance measure kdiff. **Section 2** also outlines some theoretical results for separability of data using kdiff as a distance measure for clustering, classification and related tasks by modeling the embedding distributions derived from the original data as two component mixtures. **Section 3** outlines some practical considerations and data-driven strategies to determine optimal parameters for the algorithm to estimate kdiff. **Section 4** presents simulation results using kdiff on both synthetic and real-life datasets and compares them with existing methods. Finally **Section 5** outlines some conclusions and directions for future work.

2 MAIN IDEA

2.1 Overview

Consider two real-valued datasets $\{\mathbf{X}_{\underline{t}}, \mathbf{Y}_{\underline{t}}: \underline{t} \in \mathbb{Z}^k\}$ defined over a k-dimensional index set. These may in general be vector-valued random variables, and therefore $\mathbf{X}_{\underline{t}}$ and $\mathbf{Y}_{\underline{t}}$ can be considered as either univariate or multivariate time series, random fields or other types of structured data. Our problem of interest is where instances of $\mathbf{X}_{\underline{t}}$ and $\mathbf{Y}_{\underline{t}}$ match with certain localized motifs $\{\mathbf{X}_{\underline{t}}: t \in S\} \approx \{\mathbf{Y}_{\underline{t}}: t \in S'\}$ for small localized index sets $S, S' \subset \mathbb{Z}^k$. For both the univariate and multivariate cases, we can embed these data sets into some corresponding point clouds $\mathbf{X}, \mathbf{Y} \subset \mathbb{R}^L$ via windowing with a size L window, where L can be determined from training or some other appropriate technique [8, 13]. Once we have a window embedding of these data sets, we can define various distance measures on the resulting point clouds to define similarity between $\mathbf{X}_{\underline{t}}$ and $\mathbf{Y}_{\underline{t}}$.

A distance measure that has been proposed previously to determine similarity between two such time series embedded point clouds constructed over $\mathbb{R}^L$ is MPdist [7]. In this case, a cross-data distance measure, denoted D^2, can be constructed by using 1-nearest neighbor Euclidean distances between point clouds $\mathbf{X}$ and $\mathbf{Y}$ as below:

$$\begin{aligned} d(x) &= \inf_{y \in \mathbf{Y}} \|x - y\|, \quad \forall x \in \mathbf{X} \\ d(y) &= \inf_{x \in \mathbf{X}} \|x - y\|, \quad \forall y \in \mathbf{Y} \\ D^2 &= \{d^2(z): z \in \mathbf{X} \cup \mathbf{Y}\}. \end{aligned} \tag{1}$$

In [7], the distance measure MPdist was estimated for univariate time series by choosing the kth smallest element in the set D^2. In general, MPdist can be constructed using a lower quantile of the distance distribution D^2.

Our proposed distance measure kdiff generalizes MPdist using a kernel-based construction, and by considering both cross-similarity and self-similarity. Similar to MPdist, we first construct sliding window based embeddings over the original data instances $\mathbf{X}_{\underline{t}}$, $\mathbf{Y}_{\underline{t}}$ and obtain corresponding point clouds $\mathbf{X}$ and $\mathbf{Y}$. For MPdist the final distance is estimated based on cross-similarity between the embeddings $\mathbf{X}$, $\mathbf{Y}$ as shown in **Eq. 1**. Our distance measure kdiff differs from this in two ways:

- We use a kernel based similarity measure over the obtained sliding window based embeddings $\mathbf{X}$ and $\mathbf{Y}$ for kdiff instead of the Euclidean metric used in MPdist**.**
- For kdiff the final distance is estimated based on both self and cross-similarities between the embeddings $\mathbf{X}$, $\mathbf{Y}$ respectively. The inclusion of self-similarity in the construction of kdiff as compared to only cross-similarity for MPdist leads to better clustering performance for data with reduced signal-to-noise ratio of the matching region versus the background. This is demonstrated empirically for both synthetic and real-life data in **Section 4**.

2.2 The Construction of kdiff

To define our kdff statistic, we will begin with a discussion of general distributions defined on $\mathbb{R}^L$. For the purposes of this paper, these can be assumed to be the distributions that the finite samples $\mathbf{X}$ are drawn from (in a non-iid fashion) and stitched together to form the time series $\mathbf{X}_{\underline{t}}$ (respectively for $\mathbf{Y}$ and $\mathbf{Y}_{\underline{t}}$).

In general, we can define the distributions on $\mathbb{X}$, which is a locally compact metric measure space with the metric ρ and a distinguished probability measure ν^*. The term measure will denote a signed Borel measure. We introduce a fixed, positive definite kernel $K: \mathbb{X} \times \mathbb{X} \to (0, \infty)$, $K \in C(\mathbb{X} \times \mathbb{X})$. Since the kernel is fixed, the mention of this kernel will be omitted from notations, although the kernel plays a crucial role in our theory. Given any signed measure (or positive measure having bounded variation) τ on $\mathbb{X}$, we define the witness function of τ by

$$U(\tau)(z) = \int_{\mathbb{X}} K(z,x)d\tau(x), \qquad z \in \mathbb{X}, \tag{2}$$

and similarly the magnitude of the witness function

$$T(\tau)(z) = |U(\tau)(z)|, \qquad z \in \mathbb{X}. \tag{3}$$

In the context of defining a distance between μ_1, μ_2, we take $\tau = \mu_1 - \mu_2$, which results in a witness function

$$U(\mu_1 - \mu_2)(z) = \mathbb{E}_{x\sim\mu_1}[K(z,x)] - \mathbb{E}_{y\sim\mu_2}[K(z,y)].$$

To quantify where $T\ (\mu_1 - \mu_2)$ is small, we define the cumulative distribution function (CDF) of a Borel measurable function $f\colon \mathbb{X} \to \mathbb{R}$ by

$$\Lambda(f)(t) = \Lambda(\nu^*; f)(t) = \nu^*(\{z\colon |f(z)| < t\}), \qquad t \in [0,\infty), \tag{4}$$

and its "inverse" CDF by

$$f^{\#}(u) = \sup\{t \in [0,\infty)\colon \Lambda(f)(t) \le u\}, \qquad u \in [0,\infty). \tag{5}$$

Both $\Lambda(f)$ and $f^{\#}$ are non-decreasing functions, and $f^{\#}(u)$ defines the $\boldsymbol{u}$-th quartile of f.

Finally, we are prepared to define our kdiff distance between probability measures μ_1, μ_2. Having defined $T\ (\mu_1 - \mu_2)\ (z)$, we now define kdiff to be the α quantile of $T\ (\mu_1 - \mu_2)$,

$$\mathbf{kdiff}(\mu_1, \mu_2; \alpha) = (T(\mu_1 - \mu_2))^{\#}(\alpha), \qquad \alpha \in (0,1). \tag{6}$$

The intuition of **(6)** is that, if $\mu_1 = \mu_2$, the resulting kdiff statistic will be zero. But beyond this, if $T\ (\mu_1 - \mu_2)\ (z) = 0$ for a set $z \in A \subset \mathbb{X}$ such that $\nu^*(A) > 0$, then for a localized enough kernel, there exists a quantile α for which we can still have the resulting kdiff statistic be close to zero. This allows us to match distributions that agree over partial support. This will be discussed more precisely in **Section 2.3**.

2.3 Separability Theorems for kdiff

For the purposes of analyzing the kdiff statistic, we will focus on the setting of resolving mixture models of probabilities on $\mathbb{X}$ when only one of the components agree. Accordingly, for any $\delta \in (0, 1)$, we define $\mathbb{P}_\delta$ to be the class of all probability measures μ on $\mathbb{X}$ which can be expressed as $\mu = \delta\mu_F + (1 - \delta)\mu_B$, where μ_F and μ_B are probability measures on $\mathbb{X}$. With the applications in the paper in mind, we will refer to μ_F as the *foreground* and μ_B as the *background* probabilities. Our interest is in developing a test to see whether given two measures μ_1 and μ_2 in $\mathbb{P}_\delta$, the corresponding foreground components agree. Clearly, the same discussion could also apply to the case when we wish to focus on the background components with obvious changes. We also note that in general, $\mathbb{P}_\delta$ is simply the space of probability measures, but it should be thought of as the space of mixtures of almost-disjoint measures. We will add the necessary assumptions on δ, μ_F, μ_B in the main results below.

We first present some preparatory material before reaching our desired statements. For any subset $A \subseteq \mathbb{X}$ and $x \in \mathbb{X}$, we define

$$\mathsf{dist}(A, x) = \inf_{y\in A}\rho(y, x). \tag{7}$$

The support of a finite positive measure μ, denoted by $\mathsf{supp}(\mu)$ is the set of all $x \in \mathbb{X}$ such that $\mu(U) > 0$ for all open subsets U containing x. Clearly, $\mathsf{supp}(\mu)$ is a closed set. If σ is a non-zero signed measure and $\sigma = \sigma^+ - \sigma^-$ is the Jordan decomposition of σ then we define $\mathsf{supp}(\sigma) = \mathsf{supp}(\sigma^+) \cup \mathsf{supp}(\sigma^-)$. If $f\colon \mathbb{X} \to \mathbb{R}$, we define

$$\|f\|_\infty = \sup_{x\in\mathbb{X}}|f(x)|.$$

The following lemma summarizes some important but easy properties of quantities $\Lambda(f)$ and $f^{\#}$ defined in **Eqs. 4**, **5** respectively.

Lemma 1. (a) *For* t, $u \in [0, \infty)$,

$$\Lambda(f)(t) \le u \Rightarrow t \le f^{\#}(u), \qquad u \le \Lambda(f)(t) \Rightarrow f^{\#}(u) \le t. \tag{8}$$

(b) *If* $\epsilon > 0$, $f, g\colon \mathbb{X} \to \mathbb{R}$, *and* $\|f - g\|_\infty \le \epsilon$, *then* $\sup_{u\in[0,\infty)}|f^{\#}(u) - g^{\#}(u)| \le \epsilon$.

Our goal is to investigate sufficient conditions on two measures in $\mathbb{P}_\delta$ so that kdiff can distinguish if the foreground components of the measures are the same. For this purpose, we introduce some further notation, where we suppress the mention of certain quantities for brevity. Let $\mu_j = \delta\mu_{j,F} + (1 - \delta)\mu_{j,B} \in \mathbb{P}_\delta$, $j = 1, 2$, and $\mathbb{S}_F = \mathsf{supp}(\mu_{1,F}) \cup \mathsf{supp}(\mu_{2,F})$, and we define $\mathcal{S}^c = \mathbb{X}/\mathcal{S}$. We define for η, $\theta > 0$,

$$\begin{aligned}
&\mathcal{S}_B(\mu_1, \mu_2; \eta) = \{z \in \mathbb{X}\colon T(\mu_{1,B} - \mu_{2,B})(z) < \eta\},\\
&\phi_B(\eta) = \nu^*(\mathcal{S}_B(\mu_1, \mu_2; \eta))\\
&\mathcal{S}_F(\mu_1, \mu_2; \eta) = \{z \in \mathbb{X}\colon T(\mu_{1,F} - \mu_{2,F})(z) < \eta\},\\
&\phi_F(\eta) = \nu^*(\mathcal{S}_F(\mu_1, \mu_2; \eta))\\
&\mathcal{G}(\mu_1, \mu_2; \theta, \eta) = (\mathcal{S}_F^c(\theta) \cap \mathcal{S}_B(\eta)) \cup (\mathcal{S}_B^c(\theta) \cap \mathcal{S}_F(\eta)),\\
&\psi(\theta, \eta) = \nu^*(\mathcal{G}(\mu_1, \mu_2; \theta, \eta))
\end{aligned} \tag{9}$$

Theorem 2. *Let* $\delta \in (0, \frac{1}{2})$, $\mu_j = \delta\mu_{j,F} + (1 - \delta)\mu_{j,B} \in \mathbb{P}_\delta\ (j = 1, 2)$.

a) *If* $\eta > 0$ *and* $\mu_{1,F} = \mu_{2,F}$ *then for any* $\alpha \le \phi_B(\eta)$, *we have* kdiff $(\mu_1, \mu_2; \alpha) \le (1 - \delta)\eta$.
b) *If* $\eta > 0$ *such that* $\phi_F(\frac{3(1-\delta)}{\delta}\eta) < 1$ *and* $\psi(\frac{3(1-\delta)}{\delta}\eta, \eta) > 0$, *then* $\mu_{1,F} \ne \mu_{2,F}$ *and for any* α *with*
$$1 - \psi(\frac{3(1-\delta)}{\delta}\eta, \eta) \le \alpha,$$
we have kdiff $(\mu_1, \mu_2; \alpha) \ge 2\ (1 - \delta)\eta$.

Proof. To prove part (a), we observe that since $\mu_{1,F} = \mu_{2,F}$, $T(\mu_1 - \mu_2)\ (z) = (1 - \delta) \cdot T(\mu_{1,B} - \mu_{2,B})\ (z)$ for all $z \in \mathbb{X}$. By definition (9),

$$\begin{aligned}
\mathcal{S}_B(\mu_1, \mu_2; \eta) &= \{z \in \mathbb{X}\colon T(\mu_{1,B} - \mu_{2,B})(z) < \eta\}\\
&= \{z \in \mathbb{X}\colon T(\mu_1 - \mu_2)(z) < (1 - \delta)\eta\}.
\end{aligned}$$

Therefore, $\phi_B(\eta) \le \Lambda(T(\mu_1 - \mu_2))((1-\delta)\eta)$. In view of **(8)**, this proves part (a).

To prove part (b), we will write

$$\theta = \frac{3(1-\delta)}{\delta}\eta.$$

Our hypothesis that $\phi_F(\theta) < 1$ means that $\mu_{1,F} \neq \mu_{2,F}$ and $\mathcal{S}_F^c(\theta)$ is nonempty. For all $z \in \mathcal{S}_F^c(\theta) \cap \mathcal{S}_B(\eta)$,

$$\begin{aligned} T(\mu_1 - \mu_2)(z) &\ge \delta |U(\mu_{1,F} - \mu_{2,F})(z)| - (1-\delta)|U(\mu_{1,B} - \mu_{2,B})(z)| \\ &> \delta\theta - (1-\delta)\eta \ge 2(1-\delta)\eta. \end{aligned}$$

Moreover, for $z \in \mathcal{S}_B^c(\theta) \cap \mathcal{S}_F(\eta)$, we also know that

$$\begin{aligned} T(\mu_1 - \mu_2)(z) &\ge (1-\delta)|U(\mu_{1,B} - \mu_{2,B})(z)| - \delta|U(\mu_{1,F} - \mu_{2,F})(z)| \\ &\ge (1-\delta)\theta - \delta\eta \ge \frac{3(1-\delta)^2 - \delta^2}{\delta}\eta. \end{aligned}$$

Note that because $\delta < \frac{1}{2}$, we have $\frac{3(1-\delta)^2 - \delta^2}{\delta} > 2(1-\delta)$. So, this means that

$$\{z \in \mathbb{X}\colon T(\mu_1 - \mu_2)(z) < 2(1-\delta)\eta\} \subset \{z \in \mathbb{X}\colon z \notin \mathcal{G}(\theta, \eta)\};$$

This means that $\Lambda(T(\mu_1 - \mu_2))(2(1-\delta)\eta) \le 1 - \psi(\theta, \eta)$. Since $\alpha \ge 1 - \psi(\theta, \eta)$, this estimate together with **(8)** leads to the conclusion in part (b).

We wish to comment on the practicality of the constants $\mathcal{S}_F(\mu_1, \mu_2; \eta)$, $\mathcal{S}_B(\mu_1, \mu_2; \eta)$ and $\mathcal{G}(\mu_1, \mu_2; \theta, \eta)$. We consider this with the simple setting where K is a compactly supported localized kernel (e.g., indicator function of an ϵ-ball) in order to avoid the discussion of tails. We define the well-separated setting as the setting where $\mathsf{dist}(\mathsf{supp}(\mu_{1,B}), \mathsf{supp}(\mu_{2,B})) > \epsilon$ and $\mathsf{dist}(\mathsf{supp}(\mu_{1,B}), \mathsf{supp}(\mu_{i,F})) > \epsilon$. For part (b), we'll also use $\mathsf{dist}(\mathsf{supp}(\mu_{1,F}), \mathsf{supp}(\mu_{2,F})) > \epsilon$ and all four measures are sufficiently concentrated, i.e.,

$$\begin{aligned} &\mu_{i,F}(\{z \in \mathbb{X}\colon T(\mu_{i,F})(z) \ge \theta\}) \ge 1 - \xi \\ \text{and} \quad &\mu_{i,B}(\{z \in \mathbb{X}\colon T(\mu_{i,B})(z) \ge \theta\}) \ge 1 - \xi. \end{aligned}$$

We consider the results of Theorem 2 in the well-separated setting with $\nu^* = \frac{1}{2}(\mu_1 + \mu_2)$:

a) $\mathcal{S}_B(\mu_1, \mu_2; \eta)$ measures how much the backgrounds overlap with one another. In this setting, $\phi_B(\eta) \ge \delta$ for any $\eta > 0$. This is because $T(\mu_{1,B} - \mu_{2,B})(z) = 0$ for all $z \in \mathsf{supp}(\mu_{i,F})$, and thus $z \in \mathcal{S}_B(\mu_1, \mu_2; \eta)$. Since $\nu^*(\mathsf{supp}(\mu_{1,F}) \cup \mathsf{supp}(\mu_{2,F})) = \delta$, this lower bounds $\phi_B(\eta)$. This means for any $\alpha < \delta$, kdiff $(\mu_1, \mu_2; \alpha) = 0$.

b) Because of the well-separated assumption, $\mathcal{S}_F^c(\theta) \subset \mathcal{S}_B(\eta)$. This means that everywhere the foregrounds are sufficiently concentrated, the backgrounds must be sufficiently small. Similarly, $\mathcal{S}_B^c(\theta) \subset \mathcal{S}_F(\eta)$. Furthermore, the sets $\mathcal{S}_F^c(\theta) \cap \mathcal{S}_B(\eta)$ and $\mathcal{S}_B^c(\theta) \cap \mathcal{S}_F(\eta)$ by definition are disjoint when $\theta > \eta$. Thus we have

$$\begin{aligned} 1 - \psi(\theta, \eta) &= 1 - \nu^*((\mathcal{S}_F^c(\theta) \cap \mathcal{S}_B(\eta)) \cup (\mathcal{S}_B^c(\theta) \cap \mathcal{S}_F(\eta))) \\ &= 1 - \nu^*(\mathcal{S}_F^c(\theta) \cap \mathcal{S}_B(\eta)) - \nu^*(\mathcal{S}_B^c(\theta) \cap \mathcal{S}_F(\eta)) \\ &= 1 - \nu^*(\mathcal{S}_F^c(\theta)) - \nu^*(\mathcal{S}_B^c(\theta)) \\ &\le 1 - \delta(1-\xi) - (1-\delta)(1-\xi) \\ &= \xi. \end{aligned}$$

Thus we can choose η, θ as large as possible to satisfy the assumptions, and even then for very small quantiles $\alpha > \xi$ we kdiff $(\mu_1, \mu_2; \alpha) > 2(1-\delta)\eta$.

These above descriptions clarify the theorem in the simplest setting. When the foreground distributions are small but concentrated, and far from the separate backgrounds, then the hypothesis of $\mu_{1,F} \overset{?}{=} \mu_{2,F}$ can be easily distinguished with kdiff for almost all $\alpha < \delta$.

In practice, of course, we need to estimate **kidff** $(\mu_1, \mu_2; \alpha)$ from samples taken from μ_1 and μ_2. In turn, this necessitates an estimation of the witness function of probability measure from samples from this probability. We need to do this separately for μ_1 and μ_2, but it is convenient to formulate the result for a generic probability measure μ. To estimate the error in the resulting approximation, we need to stipulate some further conditions enumerated below. We will denote by $\mathbb{S}^* = \mathsf{supp}(\mu)$.

Essential compact support For any $t > 0$, there exists $R(t) > 0$ such that

$$K(x, y) \le t, \qquad x, y \in \mathbb{X},\ \rho(x, y) \ge R(t). \tag{10}$$

Covering property For $t > 0$, let $\mathbb{B}(\mathbb{S}^*, t) = \{z \in \mathbb{X}\colon \mathsf{dist}(\mathbb{S}^*, z) \le R(t)\}$. There exist A, $\beta > 0$ such that for any $t > 0$, the set $\mathbb{B}(\mathbb{S}, t)$ is contained in the union of at most $At^{-\beta}$ balls of radius $\le t$.

Lipschitz condition We have

$$\max_{(x,y)\in\mathbb{X}\times\mathbb{X}} K(x, y) + \max_{x,x',y,y'\in\mathbb{X}} \left\{\frac{|K(x, y) - K(x', y')|}{\rho(x, x') + \rho(y, y')}\right\} \le 1. \tag{11}$$

Then Höffding's inequality leads to the following theorem.

Theorem 3. *Let $\epsilon > 0$, $M \ge 2$ be an integer, and μ be any probability measure on $\mathbb{X}$ and $\{y_1, \ldots, y_M\}$ be i.i.d. samples from μ. Then with μ-probability $\ge 1 - \epsilon$, we have*

$$\left\| U(\mu)(\circ) - \frac{1}{M}\sum_{j=1}^{M} K(\circ, y_j) \right\|_\infty \le 2\left\{\frac{\log(4^{\beta+1}A/\epsilon)}{M}\right\}^{1/2}. \tag{12}$$

The proof of Theorem 3 mirrors the results for the witness function in [14].

2.4 Conclusions From Separability Theorems

To illustrate the benefit of the above theory, we recall the MMD distance measure between two probability measures μ_1 and μ_2 defined by

$$\mathrm{MMD}^2(\mu_1,\mu_2) = \int_{\mathbb{X}}\int_{\mathbb{X}} K(x,y)\,(d\mu_1(x) - d\mu_2(x))\,(d\mu_1(y) - d\mu_2(y)). \tag{13}$$

When $\mu_1, \mu_2 \in \mathbb{P}_\delta$ and the foreground components $\mu_{1,F} = \mu_{2,F}$ then $\mu_1 - \mu_2 = (1 - \delta)$ $(\mu_{1,B}- \mu_{2,B})$ and

$$\mathrm{MMD}^2(\mu_1,\mu_2) = (1-\delta)^2\mathrm{MMD}^2(\mu_{1,B},\mu_{2,B}). \tag{14}$$

Since K is a positive definite kernel, it is thus impossible for MMD^2 $(\mu_1, \mu_2) = 0$ unless $\mu_{1,B} = \mu_{2,B}$. One of the motivations for our construction is to devise a test statistic that can be arbitrarily small even if $\mu_{1,B} \neq \mu_{2,B}$.

The results derived above provide certain insights regarding when it is possible to perform tasks such as clustering and classification of data using distance measures such as kdiff and **MMD** based on the characteristics of their foreground and background distributions. The results in Theorem 2 show that, provided the backgrounds are sufficiently separated, the kdiff statistic will be significantly smaller when $\mu_{1,F} = \mu_{2,F}$ than when $\mu_{1,F}$ and $\mu_{2,F}$ are separated.

This enables kdiff to perform accurate discrimination i.e. data belonging to the same class will be clustered correctly in this case. On the other hand, it is clear that even if $\mu_{1,F} = \mu_{2,F}$, MMD^2 (μ_1, μ_2) will still be highly dependent on the backgrounds. In this paper we consider the case where data instances belonging to the same class have the same foreground but different background distributions. In such situations using synthetic and real life examples we demonstrate the comparative performance and effectiveness of kdiff for clustering tasks versus other distance measures including MMD.

As a final note, we wish to mention the relationship between **MMD** and kdiff. It can be shown that $\mathbf{MMD}^2$ is the mean of the witness function with respect to $\nu^* = \frac{1}{2}(\mu_1+\mu_2)$, $\mathbf{MMD}^2(\mu_1,\mu_2) = \mathbb{E}_{z\sim\nu^*}(|U(\mu_1-\mu_2)(z)|^2)$ [11, 15]. This is compared to our results for kdiff, or in particular kdiff^2. Note that computing $\mathbf{kdiff}(\mu_1,\mu_2;\alpha)^2$ is equivalent to computing kdiff on the square of the witness function $T(\mu_1 - \mu_2)(z) = |U(\mu_1 - \mu_2)(z)|^2$, since quantiles depend only on the ordering of the underlying function. This means the statistic kdiff^2 is simply taking the quantile of the square of the witness function, rather than the mean as in $\mathbf{MMD}^2$.

3 ESTIMATION OF ALGORITHM PARAMETERS

The following parameters are required for estimation of the distance measure kdiff:

- Length of sliding window ***SL*** used to generate subsequences over given data (**embedding dimension**)
- Kernel bandwidth $(\boldsymbol{\sigma})$ of the Gaussian kernel $k(x,y) = e^{-\|x-y\|^2/2\sigma^2}$
- Lower quantile α of the kernel-based distance distribution $T(z)$

Determining *SL*:

In this paper we demonstrate the application of kdiff for clustering time series and random fields. The sliding window length *SL* is used to create subsequences (i.e. sliding window based embeddings) over such time series or random fields over which kdiff is estimated. The number of subsequences formed depend on *SL*, the number of points in the time series or random field and the dimensionality of the data under consideration. Some examples are given as below:

- In case of a univariate time series of length n if each subsequence is of length $L = SL$ then there are $m = n - SL + 1$ embeddings
- For a two dimensional n x n random field if each subsequence has dimension $L = SL * SL$ then there are $m = (n - SL + 1)^2$ embeddings
- For a p-variate time series if each subsequence is of length $L = p * SL$ then there are $m = n - L + 1$ embeddings

The distance measure kdiff is estimated over these m points in the L dimensional embedding space. It is necessary to determine an optimal value of *SL* to obtain accurate values of kdiff. Very small values of *SL* may result in erroneous identification of the region where the time series or random field under consideration match. For example embeddings obtained in this manner may result in two dissimilar time series containing noise related fluctuations over a small region identified as “matching”. On the other hand very high values of *SL* can lead to erroneous estimation of the distance distribution owing to less number of subsequences or sub-regions which results in incorrect estimates for kdiff. As an optimal tradeoff between these competing considerations we determine the value of *SL* based on the best clustering performance over a training set selected from the original data.

Determining σ: Since kdiff is a kernel-based similarity measure determination of the kernel bandwidth σ is critical to the accuracy of estimation. In this case very small bandwidths for sliding window based embeddings $\mathbf{X}$ and $\mathbf{Y}$ derived from two corresponding time series or random fields can lead to incorrect estimates since only points in the immediate neighborhood of embeddings $\mathbf{X}$ and $\mathbf{Y}$ are considered in the estimation of the kdiff statistic. On the other hand very large bandwidths are also problematic since in this case any point $\mathbf{Z}$ becomes nearly equidistant from $\mathbf{X}$ and $\mathbf{Y}$ (here all points are considered in the embedding space), thereby causing the distance measure to lose sensitivity. To achieve a suitable tradeoff between these extremes we select σ over a range of values of order equal to the k nearest neighbor distance over all points in the embedding space of $\mathbf{Z} = \mathbf{X} \cup \mathbf{Y}$ for a suitably chosen value of k. The optimal value of σ is selected from this range based on the best clustering performance over a training set selected from the original data.

Determining α: The distance measure kdiff is based on a lower quantile α of the estimated distance distribution over the embedding spaces of two time series or random fields. This quantile can be specified as a fraction of the total number of points in the distance distribution using either of the following methods below:

- Based on exploratory data analysis, visual inspection or other methods if the extent of the matching portions of the time series, random fields or other data under investigation can be determined then α can be set as a fraction of the length or area of this matching region versus the overall span of the data.
- For high dimensional time series or random fields α can be determined from a range of values based on the best clustering performance over a training set selected from the original data.

4 NUMERICAL WORK: SIMULATIONS AND REAL DATA

The effectiveness of the our novel distance measure kdiff for comparing two sets of data which match only partially over their region of support is estimated using kmedoids clustering [16]. The kmedoids algorithm is a similar partitional clustering algorithm as kmeans which works by minimizing the distance between the points belonging to a cluster and the center of the cluster. However kmeans can work only with Euclidean distances or a distance measure which can be directly expressed in terms of Euclidean for example the cosine distance. In contrast the kmedoids algorithm can work with non Euclidean distance measures such as kdiff and is also advantageous because the obtained cluster centers belong to one of the input data points thereby leading to greater interpretability of the results. For these reasons in this paper we consider kmedoids clustering with $k = 2$ classes and measure the accuracy of clustering for distance measures kdiff, MMD [9], MPdist [7] and dtw [1, 2] over synthetic and real time series and random field datasets as described in the following sections. Suitably chosen combinations of the parameters can be specified as described in **Section 3** and the derived optimal values can then be used for measuring clustering performance with the test data using kdiff. Similar to kdiff, distances measures using Maximum Mean Discrepancy (MMD) and MPdist are computed by first creating subsequences over the original time series or random fields. In both these cases the length of the sliding window *SL* is determined based on the best clustering performance over a training set selected from the original data. Additionally for MMD which is also a kernel-based measure we determine the optimal kernel bandwidth (σ) based on training.

In this work we consider two synthetic and one real-life datasets for measuring clustering performance with four distance measures kdiff, MMD, MPdist and dtwd (Dynamic Time Warping distance). For the synthetic datasets we generate the foregrounds and backgrounds as described in **Section 2.3** using autoregressive models of order p, denoted as AR(p). These are models for a time series W_t generated by

$$W_t = \sum_{i=1}^{p} \phi_i W_{t-i} + \epsilon_t, \qquad t = p+1, \ldots .$$

where $\phi_1, \ldots, \phi_p$ are the p coefficients of the AR(p) model and ϵ_t can be i. i.d. Gaussian errors. We perform 50 Monte Carlo runs over each dataset and in each run we randomly divide the data into training and test sets. For each set of training data we determine the optimal values of the algorithm parameters based on the best clustering performance. Following this we use these parameter values on the test data in each of the 50 runs. The final performance metric for a given distance measure is given by the total number of clustering errors for the test data over all 50 runs. The dtwclust package [17] of R 3.6.2 has been used for implementation of the kmedoids clustering algorithm and to evaluate the results of clustering.

As a techincal note, as MPdist and MMD are generally computed as squared distances, we similarly work with $\mathbf{kdiff}(\mu_1, \mu_2; \alpha)^2$ as the distance between distributions. This is solely to ensure that the distances are based of Euclidean or kernel distances squares, and to ensure a fair comparison being fed into the kmedoids clustering algorithm. Also as mentioned previously, computing $\mathbf{kdiff}(\mu_1, \mu_2; \alpha)^2$ is equivalent to computing kdiff on the square of the witness function $|U(\mu_1 - \mu_2)(z)|^2$, since quantiles depend only on the ordering of the underlying function.

4.1 Simulation: Matching Sub-regions in Univariate Time Series

Data Y_i for $i = 1, 2, \ldots, 1{,}000$ are simulated using the model (15). To generate this the series W_i are constructed via an AR (5) model driven by i. i.d errors $\sim N(0, 1)$. The AR (5) coefficients are set to 0.5, 0.1, 0.1, 0.1, 0.1.

$$Y_i = \mu + W_i \tag{15}$$

Following this we form a *background* dataset X_{B_j} by generating $j = 1, 2, \ldots, 21$ realizations of this data where the mean μ_j for realization j is set as below:

$$\mu_j = \begin{cases} 100 * j & \text{if } j \geq 1 \text{ and } j \leq 10 \\ 100 * (10 - j) & \text{if } j \geq 11 \text{ and } j \leq 20 \\ 0 & \text{if } j = 21 \end{cases}$$

Next we generate a dataset X_F consisting of 2 *foregrounds* X_{FA} and X_{FB} which enable forming the 2 classes to be considered for k-medoids clustering as follows. For foreground X_{FA} data Y_i for $i = 1, 2, \ldots, 50$ are simulated using the model (15). The series W_i is constructed via an AR (1) model driven by i. i.d errors $\sim N(0, 1)$. The AR (1) coefficient is set to 0.1 and $\mu = 10$. For foreground X_{FB} data Y_i for $i = 1, 2, \ldots, 25$ are simulated using the model (15). The series W_i is constructed via an AR (1) model driven by i. i.d errors $\sim N(0, 1)$. The AR (1) coefficient is set to 0.1 and $\mu = -10$. We then form the *foreground* dataset X_{F_j} by generating $j = 1, 2, \ldots, 21$ realizations of this data as follows:

$$X_{F_j} = \begin{cases} X_{FA} & \text{if } j \bmod 2 = 1 \\ X_{FB} & \text{if } j \bmod 2 = 0 \end{cases}$$

Finally the dataset used for clustering Z_{ij} where $i = 1, 2, \ldots, 1{,}000$ and $j = 1, 2, \ldots, 21$ is formed by mixing *backgrounds* X_B and *foregrounds* X_F as follows:

TABLE 1 | Clustering Performance for Univariate Time Series dataset with $\tau = 1$

Distance measure	Total number of errors	Percent error
kdiff	0	0
MMD	219	39.8
MPdist	0	0
Dtwd	227	41.2

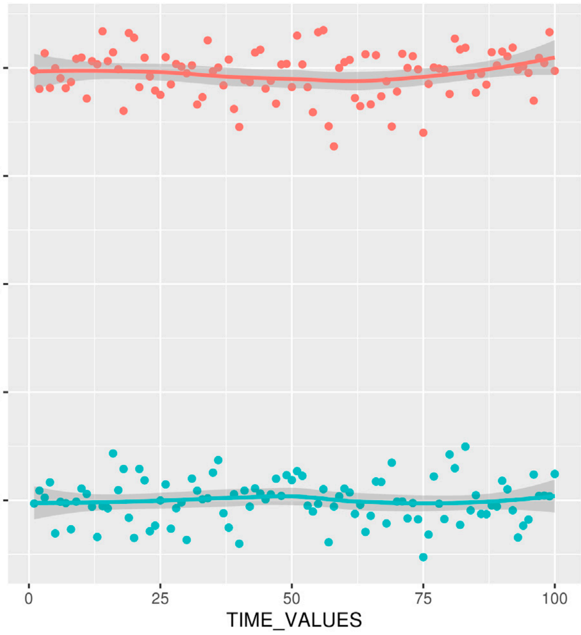

FIGURE 1 | Foreground Time Series Realization with mean = 10, − 10 for $\tau = 1$

$$
Z_{ij} = \begin{cases} \sum_{i=1}^{50} X_{F_{ij}} + \sum_{i=51}^{1000} X_{B_{ij}} & \text{if } j \bmod 2 \ = 1 \\ \sum_{i=1}^{25} X_{F_{ij}} + \sum_{i=26}^{1000} X_{B_{ij}} & \text{if } j \bmod 2 \ = 0 \end{cases}
$$

The dataset Z_{ij} formed in this manner consists of two types of subregions (*foregrounds*) which define the two classes used for k-medoids clustering. We perform 50 random splits of the dataset Z_{ij} where each split consists of a training set of size 10 and a test set of size 11. The results for clustering are shown for the 4 distance measures in **Table 1**.

From the results it can be seen that both kdiff and MPdist produce the best clustering performance with 0 errors for this dataset. This is attributed to the fact that the subregions of interest are well defined for both classes and using suitable values of parameters determined from training it is possible to accurately cluster all the time series data into two separate groups. On the other hand the performance of MMD is inferior to both kdiff and MPdist because the *backgrounds* are well separated with different mean values for time series within and across the two classes. This results in time series even belonging to the same class to be placed in separate clusters when MMD is used as a distance measure. Similarly dtwd suffers from poor performance as this distance measure tends to place time series with smaller separation between the mean background values in the same cluster. However these may have distinct values for the *foregrounds* i.e., they can in general belong to different classes and as a result this causes errors during clustering.

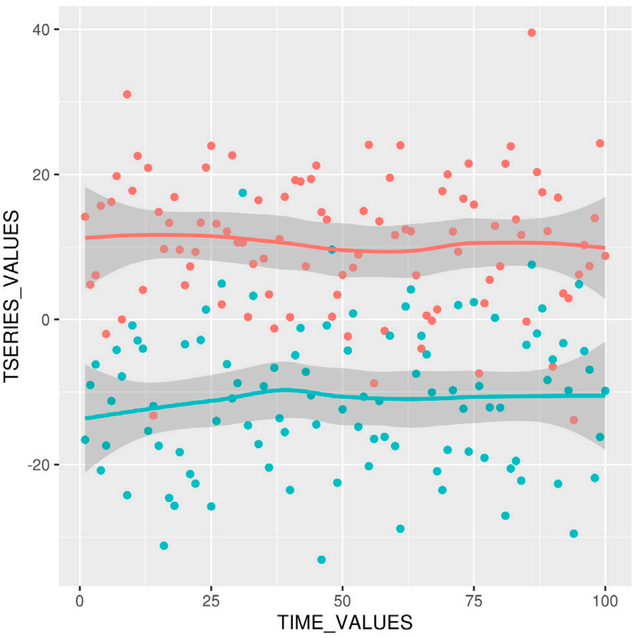

FIGURE 2 | Foreground Time Series Realizations with mean = 10, − 10 for $\tau = 10$.

TABLE 2 | Clustering Performance for Univariate Time Series dataset with $\tau = 10$

Distance measure	Total number of errors	Percent error
kdiff	20	3.6
MMD	219	39.8
MPdist	64	11.6
Dtwd	220	40.0

Noise robustness We explore the performance of the distance measures by considering noisy foregrounds. For foreground X_{FA} data Y_i for $i = 1, 2, \ldots, 50$ are simulated using the model (15). The series W_i is constructed via an AR (1) model driven by i. i.d errors $\sim N(0, 100)$. The AR (1) coefficient is set to 0.1 and $\mu = 10$. For foreground X_{FB} data Y_i for $i = 1, 2, \ldots, 25$ are simulated using the model (15). The series W_i is constructed via an AR (1) model driven by i. i.d

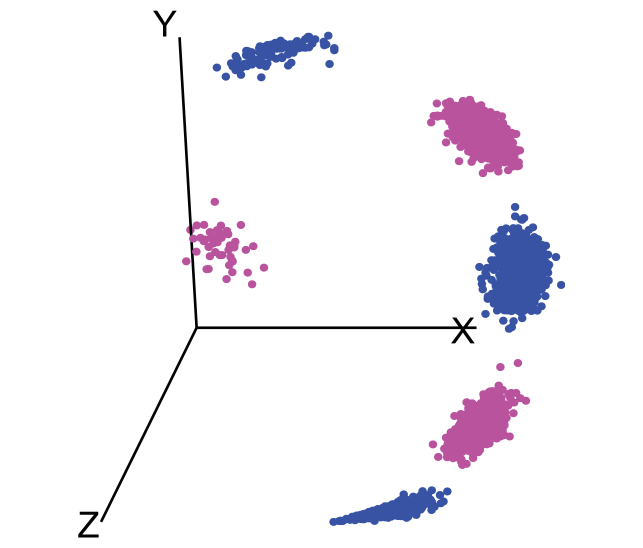

FIGURE 3 | Illustration of data consisting of 5 backgrounds and 2 foregrounds on a spherical surface, similar colors indicate association of foregrounds with respective backgrounds.

TABLE 3 | Clustering Performance for 3d Multivariate Time Series dataset.

Distance measure	Total number of errors	Percent error
kdiff	0	0
MMD	225	40.9
MPdist	141	25.6
Dtwd	227	41.3

errors $\sim N(0,1)$. The AR (1) coefficient is set to 0.1 and $\mu = -10$. Following this the foreground datasets X_{F_j} and the dataset used for clustering Z_{ij} where $i = 1, 2, \ldots, 1{,}000$ and $j = 1, 2, \ldots, 21$ are formed in the same manner as described earlier. We show example time series realizations for $\tau = 1$ and 10 in **Figures 1**, **2** respectively. Each figure contains plots of two time series with mean $= -10$, 10 as per the construction of foreground X_{FB} for the original and noisy case and show the relative separation between the realizations.

The results for clustering using these noisy foregrounds are shown in **Table 2**. The data shows empirically that as the noise level of the foreground increases kdiff is more resilient and performs better than MPdist. This is because after constructing sliding window based embeddings over the original data, MPdist is computed using Euclidean metric based cross-similarities between the embeddings whereas kdiff is estimated using kernel based self and cross similarities over the embeddings.

4.2 Simulation: Matching Sub-regions in 3-Dimensional Time Series in Spherical Coordinates

We generate a 3d multivariate *background* dataset s_B as follows. Data Y_{a_i} for $i = 1, 2, \ldots, 1{,}000$ are simulated using the model (15). To generate this the series W_i is constructed via an AR (1) model driven by i. i.d errors $\sim N(0,1)$. The AR (1) coefficient is set to 0.1. Similarly data Y_{b_i} for $i = 1, 2, \ldots, 1{,}000$ are simulated using the model (15). To generate this the series W_i is constructed via an AR (1) model driven by i. i.d errors $\sim N(0,1)$. The AR (1) coefficient is set to 0.1.

Following the generation of W_i values for the data $\mathbf{Y} = (Y_a, Y_b)$ we form a *background* dataset X_{B_j} in this 2d space by generating $j = 1, 2, \ldots, 21$ realizations of this data $\mathbf{Y}$ where the mean μ for realization j of each pair is set as below:

$$\mu = \begin{cases} 100 * j & \text{if } j \geq 1 \text{ and } j \leq 10 \\ 100 * (10 - j) & \text{if } j \geq 11 \text{ and } j \leq 20 \\ 0 & \text{j} = 21 \end{cases}$$

Our next step involves transforming these 21 instances of the 2d backgrounds into a 3d spherical surface of radius 1 as described in the following steps. We first map each series Y_a and Y_b linearly into the region $[0, \pi/2]$. The corresponding mapped series are denoted as Y_{a_s} and Y_{b_s} respectively. To ensure that the *backgrounds* are clearly separated we divide the region $[0, \pi/2]$ into 21 nonoverlapping partitions for this linear mapping. The final background dataset $s_B = \{s_a, s_b, s_c\}$ is derived using :

$$\begin{aligned} s_a &= \sin(Y_{b_s}) * \cos(Y_{a_s}) \\ s_b &= \sin(Y_{b_s}) * \sin(Y_{a_s}) \\ s_c &= \cos(Y_{b_s}) \end{aligned} \qquad (16)$$

Next we generate a 3d foreground dataset s_F consisting of 2 *foregrounds* s_{FA} and s_{FB} which will enable forming the 2 classes to be considered for k-medoids clustering as follows. Data Y_{a_i} for $i = 1, 2, \ldots, 50$ are simulated using the model (15). To generate this the series W_i is constructed via an AR (1) model driven by i. i.d errors $\sim N(0,1)$. The AR (1) coefficient is set to 0.1 and $\mu = 10$. Similarly data Y_{b_i} for $i = 1, 2, \ldots, 50$ are simulated using the model (15). To generate this the series W_i is constructed via an AR (1) model driven by i. i.d errors $\sim N(0,1)$. The AR (1) coefficient is set to 0.1 and $\mu = 10$. we linearly map the original 2d data (Y_a, Y_b) into the region $[\pi/2, 5\pi/8]$ as (Y_{a_s}, Y_{b_s}) and then perform the mapping as given in **Eq. 16** to form the *foreground* s_{FA}. The foreground s_{FB} is generated in a similar manner except that $\mu = -10$ and the 2d series is linearly mapped to the region $[3\pi/4, 7\pi/8]$. We form the *foreground* dataset s_{F_j} by generating $j = 1, 2, \ldots, 21$ realizations of this data as follows:

$$s_{F_j} = \begin{cases} s_{FA} & \text{if } j \bmod 2 = 1 \\ s_{FB} & \text{if } j \bmod 2 = 0 \end{cases}$$

Finally the dataset used for clustering Z_{ij} where $i = 1, 2, \ldots, 1{,}000$ and $j = 1, 2, \ldots, 21$ is formed by mixing *backgrounds* s_B and *foregrounds* s_F as follows:

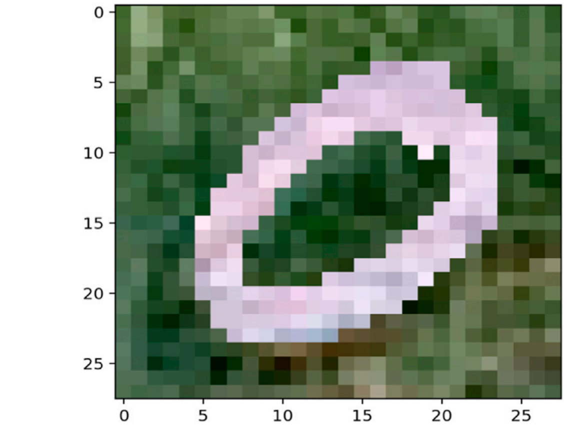

FIGURE 4 | Example 1 of MNIST-M-1 digit zero.

$$Z_{ij} = \begin{cases} \sum_{i=1}^{50} s_{F_{ij}} + \sum_{i=51}^{1000} s_{B_{ij}} & \text{if } j \bmod 2 = 1 \\ \sum_{i=1}^{25} s_{F_{ij}} + \sum_{i=26}^{1000} s_{B_{ij}} & \text{if } j \bmod 2 = 0 \end{cases}$$

The dataset Z_{ij} formed in this manner consists of two types of subregions (*foregrounds*) which define the two classes used for k-medoids clustering. An illustration of the data on such a spherical surface with 5 *backgrounds* and 2 *foregrounds* is shown in **Figure 3**.

We perform 50 random splits of the dataset Z_{ij} where each split consists of a training set of size 10 and a test set of size 11. The results for clustering are shown for the 4 distance measures in **Table 3**.

From the results it can be seen that kdiff produces the best clustering performance with 0 errors for this dataset. This is attributed to the fact that the subregions of interest are well defined for both classes and using suitable values of parameters determined from training it is possible to accurately cluster all the time series data into two separate groups. On the other hand the performance of MMD is inferior to kdiff because the backgrounds are well separated with different mean values for time series within and across the two classes. This results in time series even belonging to the same class to be placed in separate clusters when MMD is used as a distance measure. Similarly dtwd suffers from poor performance as this distance measure tends to put time series with smaller separation between the mean background values in the same cluster. However these may have distinct values for the foregrounds i.e. they can in general belong to different classes and as a result this causes errors during clustering. For this dataset the performance of MPdist is inferior to kdiff even though the former can find matching sub-regions with zero errors in the case of univariate time series. This difference is attributed to the nature of the spherical region over which the sub-region matching is done where the 1-nearest neighbor strategy employed by MPdist using Euclidean metrics to construct the distance distribution. In case

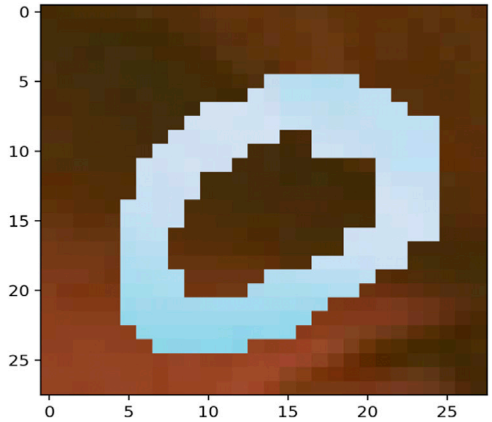

FIGURE 5 | Example 2 of MNIST-M-1 digit zero.

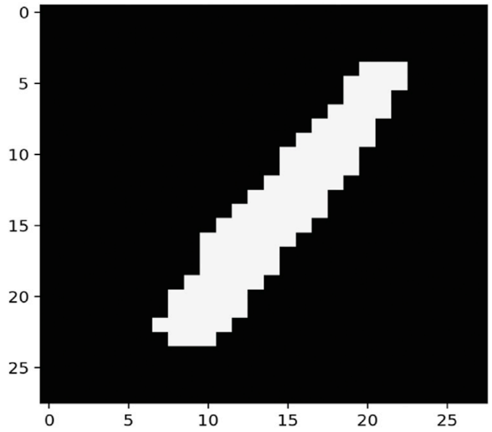

FIGURE 6 | Example 1 of MNIST-M-1 digit one.

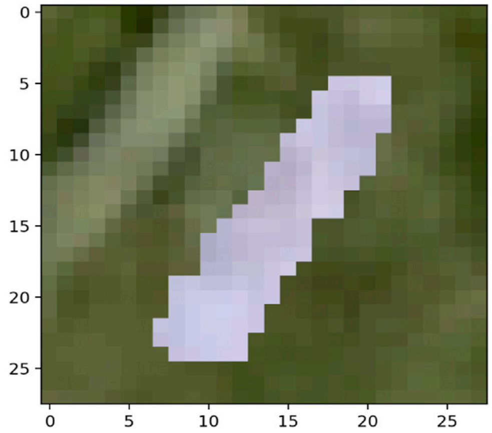

FIGURE 7 | Example 2 of MNIST-M-1 digit one.

TABLE 4 | Clustering performance for MNIST-M-1.

Distance measure	Total number of errors	Percent error
kdiff	91	18.2
MMD	131	26.2
MPdist	68	13.6
Dtwd	149	29.8

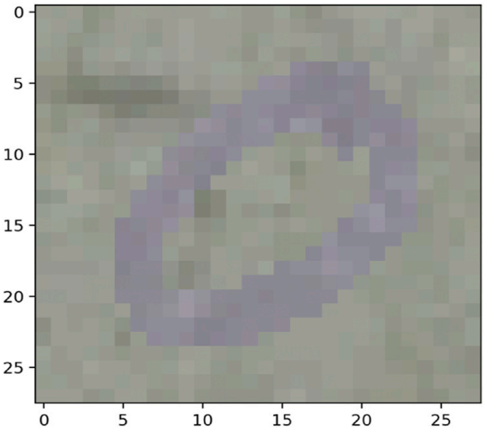

FIGURE 8 | Example 1 of MNIST-M-2 digit zero.

of spherical surfaces it is necessary to use appropriate geodesic distances for nearest neighbor search as discussed in ([18]). This issue is resolved in kdiff which can find the matching subregion over a non Euclidean region which in this case is a spherical surface thereby giving the most accurate clustering results for this dataset.

4.3 Real Life Example: MNIST-M Dataset

The MNIST-M dataset used in [19, 20] was selected as a real-life example to demonstrate the differences in clustering performance using the four distance measures kdiff, MMD, MPdist and dtwd. The MNIST-M dataset consists of MNIST digits [21] which are difference blended over patches selected from the BSDS500 database of color photos [22]. In our experiments where we consider k-medoid clustering over $k = 2$ classes we select 10 instances each of the MNIST digits 0 and 1 to be blended with a selection of background images to form our dataset MNIST-M-1. Since BSDS500 is a dataset of color images the components of this dataset are random fields whose dimensions are $28 \times 28 \times 3$. We form our final dataset for clustering consisting of random fields with dimensions 28×28 by averaging over all three channels. Examples of individual zero and one digits on different backgrounds for all three channels of MNIST-M-1 are shown in **Figures 4–7**.

We perform 50 random splits of the dataset where each split consists of a training set of size 10 and a test set of size 10. The results for clustering are shown for the distance measures in **Table 4**.

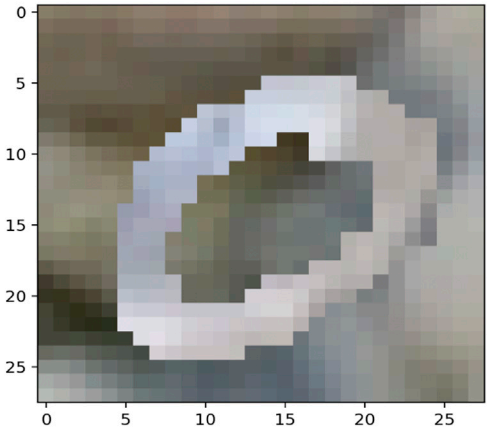

FIGURE 9 | Example 2 of MNIST-M-2 digit zero.

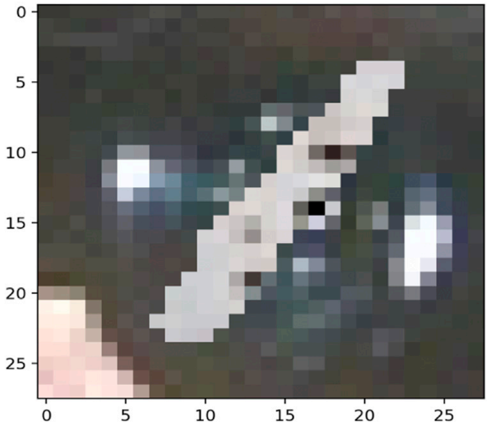

FIGURE 10 | Example 1 of MNIST-M-2 digit one.

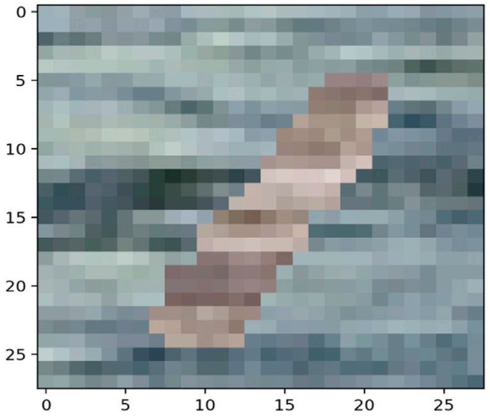

FIGURE 11 | Example 2 of MNIST-M-2 digit one.

TABLE 5 | KS statistic for MNIST-M foregrounds and backgrounds.

Dataset	KS-bg-fg-0	KS-bg-fg-1	KS-bg
MNIST-M-1	0.998	0.991	0.358
MNIST-M-2	0.889	0.754	0.202

TABLE 6 | Clustering performance for MNIST-M

Distance measure	Total number of errors	Percent error
kdiff	183	36.6
MMD	186	37.2
MPdist	197	39.4
Dtwd	197	39.4

From the results it can be seen that for MNIST-M-1 MPdist somewhat outperforms our proposed distance measure kdiff however the latter is superior to both MMD and dtwd. Since in general the background statistics of the MNIST-M images are different, two images belonging to the same class can be placed in separate clusters when MMD is used as a distance measure and this causes MMD to underperform versus kdiff. Similarly dtwd suffers from poor performance as this distance measure tends to put images with smaller separation between the mean background values in the same cluster. However these may have distinct values for the foregrounds i.e. they can in general belong to different classes and as a result this causes errors during clustering.

Noise robustness Following the discussion in **Section 4.1** we explore the performance of the distance measures by considering a selection of noisy backgrounds from the BSDS500 database over which the same 10 instances of the MNIST digits 0 and 1 are blended to form a second version of our dataset called MNIST-M-2. Similar to the earlier case we form our final dataset for clustering consisting of random fields with dimensions 28 × 28 by averaging over all three channels of the color image. Examples of individual zero and one digits on different backgrounds for a single channel are shown in **Figures 8–11**. Note that these correspond to the same MNIST digits shown in **Figures 4–7** however are blended with different backgrounds which have been chosen such that the distinguishability of the two classes is reduced.

We use the Kolmogorov-Smirnov (KS) test statistic to characterize the differences between the backgrounds (BSDS500 images) and the foregrounds (MNIST digits 0 and 1) as below:

- The mean KS statistic between the distribution of the pixels where a digit 0 is present and the distribution of the pixel values which make up the background (KS-bg-fg-0)
- The mean KS statistic between the distribution of the pixels where a digit 1 is present and the distribution of the pixel values which make up the background (KS-bg-fg-1)
- The mean KS statistic between pairs of distributions which make up the corresponding backgrounds (KS-bg)

The KS values shown in **Table 5** confirm our visual intuition that the distinguishability of the foreground (MNIST 0 and 1 digits) and the background is less for MNIST-M-2 as compared to MNIST-M-1. Additionally it can be seen that for the noisier dataset MNIST-M-2 the separation between the background distribution of pixels is less than that of MNIST-M-1.

We perform 50 random splits of the dataset Z where each split consists of a training set of size 10 and a test set of size 10. The results for clustering are shown for the distance measures in **Table 6**.

From the results it can be seen that for this noisy dataset the clustering accuracy results for all four distance measures are lower as expected, however kdiff slightly outperforms MPdist. As discussed in **Section 4.1** this can be attributed to the fact that in such cases with a lower signal to noise ratio between the foreground and the background kdiff which is estimated using kernel based self and cross similarities over the embeddings can outperform MPdist which is computed using only Euclidean metric based cross-similarities over the embeddings. The expected noise characterizaion is confirmed by our KS statistic values of KS-bg-fg-0 and KS-bg-fg-1 in **Table 6**. Moreover the lower values of the KS statistic value KS-bg for MNIST-M-2 compared to MNIST-M-1 manifest in similar clustering performances of MMD and kdiff for MNIST-M-2 in contrast with the trends for MNIST-M-1.

Additional comments: For kdiff we used $L = SL * SL$ windows for capturing the image sub-regions leading to $(n - SL + 1)^2$ embeddings which were subsequently "flattened" to form subsequences of size $L = SL^2$ over which kdiff was estimated using a one dimensional Gaussian kernel. This process can be augmented by estimating kdiff with two dimensional anisotropic Gaussian kernels to improve performance. However this augmented method of kdiff estimation using a higher dimensional kernel with more parameters will significantly increase the computation time and implementation complexity. Note that in the case of MPdist flattening the subregion is not as much of an issue since it does not use kernel based estimations which need accurate bandwidths.

5 CONCLUSIONS AND FUTURE WORK

In this work we have proposed a kernel-based measure kdiff for estimating distances between time series, random fields and similar univariate or multivariate and possibly non-iid data. Such a distance measure can be used for clustering and classification in applications where data belonging to a given class match only partially over their region of support. In such cases kdiff is shown to outperform both Maximum Mean Discrepancy and Dynamic Time Warping based distance measures for both synthetic and real-life datasets. We also compare the performance of kdiff which is constructed using kernel-based embeddings over the given data versus MPdist which uses Euclidean distance based embeddings. In this case we empirically demonstrate that for data with high signal-to-noise ratio between the matching region and the background both kdiff and MPdist perform equally well for synthetic datasets and MPdist somewhat outperforms kdiff for real life MNIST-M data. For data where the foreground is less distinguishable versus the background kdiff outperforms MPdist for both synthetic and real-life datasets. Additionally for multivariate time series on a spherical manifold we show that kdiff outperforms MPdist because of its kernel-based construction which leads to superior performance in non Euclidean spaces. Our future work will focus on application of kdiff for applications on spherical manifolds such as text embedding [23] and hyperspectral imagery [18, 24] as well as clustering and classification applications for time series and random fields with noisy motifs and foregrounds.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: Ganin et al., 2016 [19].

AUTHOR CONTRIBUTIONS

SD and AC contributed to conception and design. All authors contributed to the theory. SD performed the computation and analysis, and wrote the first draft of the manuscript. All authors contributed to the manuscript revision.

FUNDING

This work was supported in part by the NSF awards CNS1730158, ACI-1540112, ACI-1541349, OAC-1826967, the University of California Office of the President, and the California Institute for Telecommunications and Information Technology's Qualcomm Institute (Calit2-QI). SD was partially supported by an Intel Sponsored Research grant. AC was supported by NSF awards 1819222, 2012266, Russel Sage Founation 2196, and an Intel Sponsored Research grant. HM is supported in part by NSF grant DMS 2012355 and ARO Grant W911NF2110218.

ACKNOWLEDGMENTS

Thanks to CENIC for the 100Gpbs networks. The authors thank Siqiao Ruan for helpful discussions.

REFERENCES

1. Ratanamahatana CA, Keogh E. Everything You Know about Dynamic Time Warping Is Wrong. In: Third Workshop on Mining Temporal and Sequential Data, Vol. 32, Seattle. Citeseer (2004).
2. Keogh E, Ratanamahatana CA. Exact Indexing of Dynamic Time Warping. *Knowl Inf Syst* (2005) 7(3):358–86. doi:10.1007/s10115-004-0154-9
3. D'Urso P, Maharaj EA. Autocorrelation-based Fuzzy Clustering of Time Series. *Fuzzy Sets Syst* (2009) 160(24):3565–89.
4. Golay X, Kollias S, Stoll G, Meier D, Valavanis A, Boesiger P. A New Correlation-Based Fuzzy Logic Clustering Algorithm for Fmri. *Magn Reson Med* (1998) 40(2):249–60. doi:10.1002/mrm.1910400211
5. Alonso AM, Casado D, López-Pintado S, Romo J. Robust Functional Supervised Classification for Time Series. *J Classif* (2014) 31(3):325–50. doi:10.1007/s00357-014-9163-x
6. D'Urso P, Maharaj EA, Alonso AM. Fuzzy Clustering of Time Series Using Extremes. *Fuzzy Sets Syst* (2017) 318:56–79.
7. Gharghabi S, Imani S, Bagnall A, Darvishzadeh A, Keogh E. Matrix Profile Xii: Mpdist: A Novel Time Series Distance Measure to Allow Data Mining in More Challenging Scenarios. In: 2018 IEEE International Conference on Data Mining (ICDM). IEEE (2018). p. 965–70. doi:10.1109/icdm.2018.00119
8. Brandmaier AM. *Permutation Distribution Clustering and Structural Equation Model Trees*. CRAN (2011).
9. Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A. A Kernel Two-Sample Test. *J Machine Learn Res* (2012) 13:723–73.
10. Gretton A, Sejdinovic D, Strathmann H, Balakrishnan S, Pontil M, Fukumizu K, et al. Optimal Kernel Choice for Large-Scale Two-Sample Tests. In: *Advances in Neural Information Processing Systems*. Citeseer (2012). p. 1205–13.
11. Szabo Z, Chwialkowski K, Gretton A, Jitkrittum W. Interpretable Distribution Features with Maximum Testing Power. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. IEEE (2016).
12. Cheng X, Cloninger A, Coifman RR. Two-sample Statistics Based on Anisotropic Kernels. *Inf Inference: A J IMA* (2020) 9(3):677–719. doi:10.1093/imaiai/iaz018
13. Bandt C, Pompe B. Permutation Entropy: a Natural Complexity Measure for Time Series. *Phys Rev Lett* (2002) 88(17):174102. doi:10.1103/physrevlett.88.174102
14. Mhaskar HN, Cheng X, Cloninger A. A Witness Function Based Construction of Discriminative Models Using Hermite Polynomials. *Front Appl Math Stat* (2020) 6:31. doi:10.3389/fams.2020.00031
15. Cloninger A. *Bounding the Error from Reference Set Kernel Maximum Mean Discrepancy*. arXiv preprint arXiv:1812.04594 (2018).
16. Kaufmann L, Rousseeuw P. Clustering by Means of Medoids. In: Proc. Statistical Data Analysis Based on the L1 Norm Conference. Neuchatel (19871987). p. 405–16.
17. Sardá-Espinosa A. Comparing Time-Series Clustering Algorithms in R Using the Dtwclust Package. *R Package Vignette* (2017) 12:41.
18. Lunga D, Ersoy O. Spherical Nearest Neighbor Classification: Application to Hyperspectral Data. In: International Workshop on Machine Learning and Data Mining in Pattern Recognition. Springer (2011). p. 170–84. doi:10.1007/978-3-642-23199-5_13
19. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, et al. Domain-adversarial training of neural networks, *J Machine Learn Res* 17(1), pp. 2096 (2030, 2016).
20. Haeusser P, Frerix T, Mordvintsev A, Cremers D. Associative Domain Adaptation. In: Proceedings of the IEEE International Conference on Computer Vision. IEEE (2017). p. 2765–73. doi:10.1109/iccv.2017.301
21. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based Learning Applied to Document Recognition. *Proc IEEE* (1998) 86(11):2278–324. doi:10.1109/5.726791
22. Arbeláez P, Maire M, Fowlkes C, Malik J. Contour Detection and Hierarchical Image Segmentation. *IEEE Trans Pattern Anal Mach Intell* (2010) 33(5): 898–916. doi:10.1109/TPAMI.2010.161
23. Meng Y, Huang J, Wang G, Zhang C, Zhuang H, Kaplan L, et al. Spherical Text Embedding. In: *Advances in Neural Information Processing Systems*. Springer (2019). p. 8208–17.
24. Lunga D, Ersoy O. Unsupervised Classification of Hyperspectral Images on Spherical Manifolds. In: Industrial Conference on Data Mining. Springer (2011). p. 134–46. doi:10.1007/978-3-642-23184-1_11

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.