UNIVERSITÄT SALZBURG

A Stable Multi-Scale Kernel for Topological Machine Learning

Jan Reininghaus Stefan Huber Ulrich Bauer Roland Kwitt

Technical Report 2014-09 December 2014

Department of Computer Sciences

Jakob-Haringer-Straße 2
5020 Salzburg
Austria
www.cosy.sbg.ac.at

Technical Report Series

A Stable Multi-Scale Kernel for Topological Machine Learning

Jan Reininghaus, Stefan Huber
IST Austria

Ulrich Bauer
IST Austria, TU München

Roland Kwitt
University of Salzburg, Austria

Abstract

Topological data analysis offers a rich source of valuable information to study vision problems. Yet, so far we lack a theoretically sound connection to popular kernel-based learning techniques, such as kernel SVMs or kernel PCA. In this work, we establish such a connection by designing a multi-scale kernel for persistence diagrams, a stable summary representation of topological features in data. We show that this kernel is positive definite and prove its stability with respect to the 1-Wasserstein distance. Experiments on two benchmark datasets for 3D shape classification/retrieval and texture recognition show considerable performance gains of the proposed method compared to an alternative approach that is based on the recently introduced persistence landscapes.

1. Introduction

In many computer vision problems, data (*e.g.*, images, meshes, point clouds, etc.) is piped through complex processing chains in order to extract information that can be used to address high-level inference tasks, such as recognition, detection or segmentation. The extracted information might be in the form of low-level appearance descriptors, *e.g.*, SIFT [21], or of higher-level nature, *e.g.*, activations at specific layers of deep convolutional networks [19]. In recognition problems, for instance, it is then customary to feed the consolidated data to a discriminant classifier such as the popular support vector machine (SVM), a kernel-based learning technique.

While there has been substantial progress on extracting and encoding discriminative information, only recently have people started looking into the *topological structure* of the data as an additional source of information. With the emergence of *topological data analysis (TDA)* [6], computational tools for efficiently identifying topological structure have become readily available. Since then, several authors have demonstrated that TDA can capture characteristics of the data that other methods often fail to provide, *c.f.* [28, 20].

Along these lines, studying persistent homology [13] is a particularly popular method for TDA, since it captures the birth and death times of topological features, *e.g.*, connected components, holes, etc., at multiple scales. This information is summarized by the *persistence diagram*, a multiset of points in the plane. The key feature of persistent homology is its stability: small changes in the input data lead to small changes in the Wasserstein distance of the associated persistence diagrams [12]. Considering the discrete nature of topological information, the existence of such a well-behaved summary is perhaps surprising.

Note that persistence diagrams together with the Wasserstein distance only form a metric space. Thus it is not possible to directly employ persistent homology in the large class of machine learning techniques that require a Hilbert space structure, like SVM or PCA. This obstacle is typically circumvented by defining a kernel function on the domain containing the data, which in turn defines a Hilbert space structure implicitly. While the Wasserstein distance itself does not naturally lead to a valid kernel (see Appendix A), we show that it is possible to define a kernel for persistence diagrams that is stable w.r.t. the 1-Wasserstein distance. This is the main contribution of this paper.

Contribution. We propose a (positive definite) multi-scale kernel for persistence diagrams (see Fig. 1). This kernel is defined via an L_2-valued feature map, based on ideas from scale space theory [17]. We show that our feature map is Lipschitz continuous with respect to the 1-Wasserstein distance, thereby maintaining the stability property of persistent homology. The scale parameter of our kernel controls its robustness to noise and can be tuned to the data. We investigate, in detail, the theoretical properties of the kernel, and demonstrate its applicability on shape classification/retrieval and texture recognition benchmarks.

2. Related work

Methods that leverage topological information for computer vision or medical imaging methods can roughly be grouped into two categories. In the first category, we identify previous work that *directly* utilizes topological information to address a specific problem, such as topology-guided segmentation. In the second category, we identify approaches that *indirectly* use topological information. That

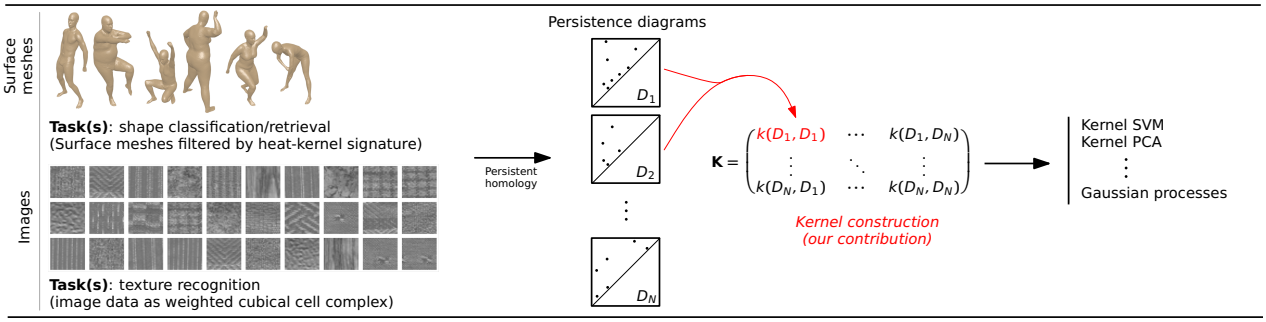

Figure 1: Visual data (*e.g.*, functions on surface meshes, textures, etc.) is analyzed using persistent homology [13]. Roughly speaking, persistent homology captures the birth/death times of topological features (*e.g.*, connected components or holes) in the form of *persistence diagrams*. Our contribution is to define a *kernel for persistence diagrams* to enable a theoretically sound use these summary representations in the framework of kernel-based learning techniques, popular in the computer vision community.

is, information about topological features is used as input to some machine-learning algorithm.

As a representative of the first category, Skraba *et al.* [28] adapt the idea of persistence-based clustering [8] in a segmentation method for surface meshes of 3D shapes, driven by the topological information in the persistence diagram. Gao *et al.* [14] use persistence information to restore so called *handles*, *i.e.*, topological cycles, in already existing segmentations of the left ventricle, extracted from computed tomography images. In a different segmentation setup, Chen *et al.* [9] propose to directly incorporate topological constraints into random-field based segmentation models.

In the second category of approaches, Chung *et al.* [10] and Pachauri *et al.* [23] investigate the problem of analyzing cortical thickness measurements on 3D surface meshes of the human cortex in order to study developmental and neurological disorders. In contrast to [28], persistence information is not used directly, but rather as a *descriptor* that is fed to a discriminant classifier in order to distinguish between normal control patients and patients with Alzheimer's disease/autism. Yet, the step of training the classifier with topological information is typically done in a rather adhoc manner. In [23] for instance, the persistence diagram is first rasterized on a regular grid, then a kernel-density estimate is computed, and eventually the vectorized discrete probability density function is used as a feature vector to train a SVM using standard kernels for $\mathbb{R}^n$. It is however unclear how the resulting kernel-induced distance behaves with respect to existing metrics (*e.g.*, bottleneck or Wasserstein distance) and how properties such as stability are affected. An approach that directly uses well-established distances between persistence diagrams for recognition was recently proposed by Li *et al.* [20]. Besides bottleneck and Wasserstein distance, the authors employ persistence landscapes [5] and the corresponding distance in their experiments. Their results expose the complementary nature of persistence information when combined with traditional bag-of-feature approaches. While our empirical study in Sec. 5.2 is inspired by [20], we primarily focus on the development of the kernel; the combination with other methods is straightforward.

In order to enable the use of persistence information in machine learning setups, Adcock *et al.* [1] propose to compare persistence diagrams using a feature vector motivated by algebraic geometry and invariant theory. The features are defined using algebraic functions of the birth and death values in the persistence diagram.

From a conceptual point of view, Bubenik's concept of *persistence landscapes* [5] is probably the closest to ours, being another kind of feature map for persistence diagrams. While persistence landscapes were not explicitly designed for use in machine learning algorithms, we will draw the connection to our work in Sec. 5.1 and show that they in fact admit the definition of a valid positive definite kernel. Moreover, both persistence landscapes as well as our approach represent computationally attractive alternatives to the bottleneck or Wasserstein distance, which both require the solution of a matching problem.

3. Background

First, we review some fundamental notions and results from persistent homology that will be relevant for our work.

Persistence diagrams. *Persistence diagrams* are a concise description of the topological changes occuring in a growing sequence of shapes, called *filtration*. In particular, during the growth of a shape, holes of different dimension (*i.e.*, gaps between components, tunnels, voids, etc.) may appear and disappear. Intuitively, a k-dimensional hole, born at time b and filled at time d, gives rise to a point (b, d) in the k^{th} persistence diagram. A persistence diagram is thus a multiset of points in $\mathbb{R}^2$. Formally, the persistence diagram

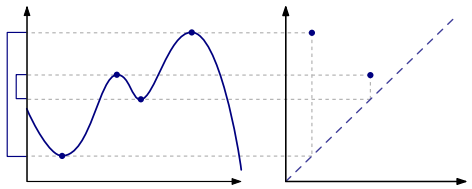

Figure 2: A function $\mathbb{R} \to \mathbb{R}$ (left) and its 0^{th} persistence diagram (right). Local minima create a connected component in the corresponding sublevel set, while local maxima merge connected components. The pairing of birth and death is shown in the persistence diagram.

is defined using a standard concept from algebraic topology called *homology*; see [13] for details.

Note that not every hole has to disappear in a filtration. Such holes give rise to *essential* features and are naturally represented by points of the form (b, ∞) in the diagram. Essential features therefore capture the topology of the final shape in the filtration. In the present work, we do not consider these features as part of the persistence diagram. Moreover, all persistence diagrams will be assumed to be finite, as is usually the case for persistence diagrams coming from data.

Filtrations from functions. A standard way of obtaining a filtration is to consider the *sublevel sets* $f^{-1}(-\infty, t]$ of a function $f\colon \Omega \to \mathbb{R}$ defined on some domain Ω, for $t \in \mathbb{R}$. It is easy to see that the sublevel sets indeed form a filtration parametrized by t. We denote the resulting persistence diagram by D_f; see Fig. 2 for an illustration.

As an example, consider a grayscale image, where Ω is the rectangular domain of the image and f is the grayscale value at any point of the domain (*i.e.*, at a particular pixel). A sublevel set would thus consist of all pixels of Ω with value up to a certain threshold t. Another example would be a piecewise linear function on a triangular mesh Ω, such as the popular heat kernel signature [29]. Yet another commonly used filtration arises from point clouds P embedded in $\mathbb{R}^n$, by considering the distance function $d_P(x) = \min_{p \in P} \|x - p\|$ on $\Omega = \mathbb{R}^n$. The sublevel sets of this function are unions of balls around P. Computationally, they are usually replaced by equivalent constructions called *alpha shapes*.

Stability. A crucial aspect of the persistence diagram D_f of a function f is its stability with respect to perturbations of f. In fact, only stability guarantees that one can infer information about the function f from its persistence diagram D_f in the presence of noise.

Formally, we consider $f \mapsto D_f$ as a map of metric spaces and define *stability* as Lipschitz continuity of this map. This requires choices of metrics both on the set of functions and the set of persistence diagrams. For the functions, the L_∞ metric is commonly used.

There is a natural metric associated to persistence diagrams, called the *bottleneck distance*. Loosely speaking, the distance of two diagrams is expressed by minimizing the largest distance of any two corresponding points, over all bijections between the two diagrams. Formally, let F and G be two persistence diagrams, each augmented by adding each point (t, t) on the diagonal with countably infinite multiplicity. The *bottleneck distance* is

$$d_B(F, G) = \inf_{\gamma} \sup_{x \in F} \|x - \gamma(x)\|_\infty, \tag{1}$$

where γ ranges over all bijections from the individual points of F to the individual points of G. As shown by Cohen-Steiner *et al.* [11], persistence diagrams are stable with respect to the bottleneck distance.

The bottleneck distance embeds into a more general class of distances, called *Wasserstein distances*. For any positive real number p, the *p-Wasserstein distance* is

$$d_{W,p}(F, G) = \left(\inf_{\gamma} \sum_{x \in F} \|x - \gamma(x)\|_\infty^p \right)^{\frac{1}{p}}, \tag{2}$$

where again γ ranges over all bijections from the individual elements of F to the individual elements of G. Note that taking the limit $p \to \infty$ yields the bottleneck distance, and we therefore define $d_{W,\infty} = d_B$. We have the following result bounding the p-Wasserstein distance in terms of the L_∞ distance:

Theorem 1 (Cohen-Steiner et al. [12])**.** *Assume that X is a compact triangulable metric space such that for every 1-Lipschitz function f on X and for $k \geq 1$, the* degree k total persistence $\sum_{(b,d) \in D_f} (d-b)^k$ *is bounded above by some constant C. Let f, g be two L-Lipschitz piecewise linear functions on X. Then for all $p \geq k$,*

$$d_{W,p}(D_f, D_g) \leq (LC)^{\frac{1}{p}} \|f - g\|_\infty^{1 - \frac{k}{p}}. \tag{3}$$

We note that, strictly speaking, this is not a stability result in the sense of Lipschitz continuity, since it only establishes Hölder continuity. Moreover, it only gives a constant upper bound for the Wasserstein distance when $p = 1$.

Kernels. Given a set $\mathcal{X}$, a function $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a *kernel* if there exists a Hilbert space $\mathcal{H}$, called *feature space*, and a map $\Phi\colon \mathcal{X} \to \mathcal{H}$, called *feature map*, such that $k(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$ for all $x, y \in \mathcal{X}$. Equivalently, k is a kernel if it is symmetric and positive definite [26]. Kernels allow to apply machine learning algorithms operating on a Hilbert space to be applied to more general settings, such as strings, graphs, or, in our case, persistence diagrams.

A kernel induces a pseudometric $d_k(x,y) = (k(x,x) + k(y,y) - 2\,k(x,y))^{1/2}$ on $\mathcal{X}$, which is the distance $\|\Phi(x) - \Phi(y)\|_{\mathcal{H}}$ in the feature space. We call the kernel k *stable* w.r.t. a metric d on $\mathcal{X}$ if there is a constant $C > 0$ such that $d_k(x,y) \leq C\,d(x,y)$ for all $x, y \in \mathcal{X}$. Note that this is equivalent to Lipschitz continuity of the feature map.

The stability of a kernel is particularly useful for classification problems: assume that there exists a separating hyperplane H for two classes of data points with margin m. If the data points are perturbed by some $\epsilon < m/2$, then H still separates the two classes with a margin $m - 2\epsilon$.

4. The persistence scale-space kernel

We propose a stable *multi-scale* kernel k_σ for the set of persistence diagrams $\mathcal{D}$. This kernel will be defined via a feature map $\Phi_\sigma : \mathcal{D} \to L_2(\Omega)$, with $\Omega \subset \mathbb{R}^2$ denoting the closed half plane above the diagonal.

To motivate the definition of Φ_σ, we point out that the set of persistence diagrams, *i.e.*, multisets of points in $\mathbb{R}^2$, does not possess a Hilbert space structure per se. However, a persistence diagram D can be uniquely represented as a sum of Dirac delta distributions[1], one for each point in D. Since Dirac deltas are functionals in the Hilbert space $H^{-2}(\mathbb{R}^2)$ [18, Chapter 7], we obtain a canonical Hilbert space structure for persistence diagrams by adopting this point of view.

Unfortunately, the induced metric on $\mathcal{D}$ does *not* take into account the distance of the points in the diagrams or to the diagonal, and therefore cannot be robust against perturbations of the diagrams. Motivated by scale-space theory [17], we address this issue by using the sum of Dirac deltas as an initial condition for a heat diffusion problem with a Dirichlet boundary condition on the diagonal. The solution of this partial differential equation is an $L_2(\Omega)$ function for any chosen scale parameter $\sigma > 0$. In the following paragraphs, we will

1) define the persistence scale space kernel k_σ,
2) derive a simple formula for evaluating k_σ, and
3) prove stability of k_σ w.r.t. the 1-Wasserstein distance.

Definition 1. *Let $\Omega = \{x = (x_1, x_2) \in \mathbb{R}^2 \colon x_2 \geq x_1\}$ denote the space above the diagonal, and let δ_p denote a Dirac delta centered at the point p. For a given persistence diagram D, we now consider the solution $u \colon \Omega \times \mathbb{R}_{\geq 0} \to \mathbb{R}, (x,t) \mapsto u(x,t)$ of the partial differential equation*[2]

$$\Delta_x u = \partial_t u \qquad \text{in } \Omega \times \mathbb{R}_{>0}, \tag{4}$$

$$u = 0 \qquad \text{on } \partial\Omega \times \mathbb{R}_{\geq 0}, \tag{5}$$

$$u = \sum_{p \in D} \delta_p \qquad \text{on } \Omega \times \{0\}. \tag{6}$$

[1] A Dirac delta distribution is a functional that evaluates a given smooth function at a point.

[2] Since the initial condition (13) is not an $L_2(\Omega)$ function, this equation is to be understood in the sense of distributions. For a rigorous treatment of existence and uniqueness of the solution, see [18, Chapter 7].

The feature map $\Phi_\sigma \colon \mathcal{D} \to L_2(\Omega)$ at scale $\sigma > 0$ of a persistence diagram D is now defined as $\Phi_\sigma(D) = u|_{t=\sigma}$. This map yields the persistence scale space kernel k_σ on $\mathcal{D}$ as

$$k_\sigma(F,G) = \langle \Phi_\sigma(F), \Phi_\sigma(G) \rangle_{L_2(\Omega)}. \tag{7}$$

Note that $\Phi_\sigma(D) = 0$ for some $\sigma > 0$ implies that $u = 0$ on $\Omega \times \{0\}$, which means that D has to be the empty diagram. From linearity of the solution operator it now follows that Φ_σ is an injective map.

The solution of the partial differential equation can be obtained by extending the domain from Ω to $\mathbb{R}^2$ and replacing (13) with

$$u = \sum_{p \in D} \delta_p - \delta_{\overline{p}} \qquad \text{on } \mathbb{R}^2 \times \{0\}, \tag{8}$$

where $\overline{p} = (b,a)$ is $p = (a,b)$ mirrored at the diagonal. It can be shown that restricting the solution of this extended problem to Ω yields a solution for the original equation. It is given by convolving the initial condition (8) with a Gaussian kernel:

$$u(x,t) = \frac{1}{4\pi t} \sum_{p \in D} e^{-\frac{\|x-p\|^2}{4t}} - e^{-\frac{\|x-\overline{p}\|^2}{4t}}. \tag{9}$$

Using this closed form solution of u, we can derive a simple expression for evaluating the kernel explicitly:

$$k_\sigma(F,G) = \frac{1}{8\pi\sigma} \sum_{\substack{p \in F \\ q \in G}} e^{-\frac{\|p-q\|^2}{8\sigma}} - e^{-\frac{\|p-\overline{q}\|^2}{8\sigma}}. \tag{10}$$

We refer to Appendix C for the elementary derivation of (10) and for a visualization (see Appendix B) of the solution (9). Note that the kernel can be computed in $O(|F| \cdot |G|)$ time, where $|F|$ and $|G|$ denote the cardinality of the multisets F and G, respectively.

Theorem 2. *The kernel k_σ is 1-Wasserstein stable.*

Proof. To prove 1-Wasserstein stability of k_σ, we show Lipschitz continuity of the feature map Φ_σ as follows:

$$\|\Phi_\sigma(F) - \Phi_\sigma(G)\|_{L_2(\Omega)} \leq \frac{1}{\sigma\sqrt{8\pi}}\, d_{W,1}(F,G), \tag{11}$$

where F and G denote persistence diagrams that have been augmented with points on the diagonal. Note that augmenting diagrams with points on the diagonal does not change the values of Φ_σ, as can be seen from (9). Since the unaugmented persistence diagrams are assumed to be finite, some matching γ between F and G achieves the infimum in the definition of the Wasserstein distance, $d_{W,1}(F,G) = \sum_{u \in F} \|u - \gamma(u)\|_\infty$. Writing $N_u(x) = \frac{1}{4\pi\sigma} e^{-\frac{\|x-u\|_2^2}{4\sigma}}$, we have

$\|N_u - N_v\|_{L_2(\mathbb{R}^2)} = \frac{1}{\sqrt{4\pi\sigma}} \cdot \sqrt{1 - e^{-\frac{\|u-v\|_2^2}{8\sigma}}}$. The Minkowski inequality and the inequality $e^{-\xi} \geq 1 - \xi$ finally yield

$$
\begin{aligned}
&\|\Phi_\sigma(F) - \Phi_\sigma(G)\|_{L_2(\Omega)} \\
&\leq \left\| \sum_{u \in F} (N_u - N_{\overline{u}}) - (N_{\gamma(u)} - N_{\overline{\gamma(u)}}) \right\|_{L_2(\mathbb{R}^2)} \\
&\leq 2 \sum_{u \in F} \|N_u - N_{\gamma(u)}\|_{L_2(\mathbb{R}^2)} \\
&\leq \frac{1}{\sqrt{\pi\sigma}} \sum_{u \in F} \sqrt{1 - e^{-\frac{\|u-\gamma(u)\|_2^2}{8\sigma}}} \\
&\leq \frac{1}{\sigma\sqrt{8\pi}} \sum_{u \in F} \|u - \gamma(u)\|_2 \quad \leq \quad \frac{1}{2\sigma\sqrt{\pi}} d_{W,1}(F,G). \qquad \square
\end{aligned}
$$

We refer to the left-hand side of (11) as the *persistence scale space distance* d_{k_σ} between F and G. Note that the right hand side of (11) decreases as σ increases. Adjusting σ accordingly allows to counteract the influence of noise in the input data, which causes an increase in $d_{W,1}(F,G)$. We will see in Sec. 5.3 that tuning σ to the data can be beneficial for the overall performance of machine learning methods.

A natural question arising from Theorem 2 is whether our stability result extends to $p > 1$. To answer this question, we first note that our kernel is *additive:* we call a kernel k on persistence diagrams additive if $k(E \cup F, G) = k(E,G) + k(F,G)$ for all $E, F, G \in \mathcal{D}$. By choosing $F = \emptyset$, we see that if k is additive then $k(\emptyset, G) = 0$ for all $G \in \mathcal{D}$. We further say that a kernel k is *trivial* if $k(F,G) = 0$ for all $F, G \in \mathcal{D}$. The next theorem establishes that Theorem 2 is sharp in the sense that *no* non-trivial additive kernel can be stable w.r.t. the p-Wasserstein distance when $p > 1$.

Theorem 3. *A non-trivial additive kernel k on persistence diagrams is not stable w.r.t. $d_{W,p}$ for any $1 < p \leq \infty$.*

Proof. By the non-triviality of k, it can be shown that there exists an $F \in \mathcal{D}$ such that $k(F,F) > 0$. We prove the claim by comparing the rates of growth of $d_{k_\sigma}(\bigcup_{i=1}^n F, \emptyset)$ and $d_{W,p}(\bigcup_{i=1}^n F, \emptyset)$ w.r.t. n. We have

$$
d_{k_\sigma}\left(\bigcup_{i=1}^{n} F, \emptyset \right) = n \sqrt{k(F,F)}.
$$

On the other hand,

$$
d_{W,p}\left(\bigcup_{i=1}^{n} F, \emptyset \right) = d_{W,p}(F, \emptyset) \cdot \begin{cases} \sqrt[p]{n} & \text{if } p < \infty, \\ 1 & \text{if } p = \infty. \end{cases}
$$

Hence, d_{k_σ} can not be bounded by $C \cdot d_{W,p}$ with a constant $C > 0$ if $p > 1$. $\square$

5. Evaluation

To evaluate the kernel proposed in Sec. 4, we investigate conceptual differences to persistence landscapes in Sec. 5.1, and then consider its performance in the context of shape classification/retrieval and texture recognition in Sec. 5.2.

5.1. Comparison to persistence landscapes

In [5], Bubenik introduced *persistence landscapes*, a representation of persistence diagrams as functions in the Banach space $L_p(\mathbb{R}^2)$. This construction was mainly intended for statistical computations, enabled by the vector space structure of L_p. For $p = 2$, we can use the Hilbert space structure of $L_2(\mathbb{R}^2)$ to construct a kernel analogously to (7). For the purpose of this work, we refer to this kernel as the *persistence landscape kernel* k^L and denote by $\Phi^L \colon \mathcal{D} \to L_2(\mathbb{R}^2)$ the corresponding feature map. The kernel-induced distance is denoted by d_{k^L}. Bubenik shows stability w.r.t. a weighted version of the Wasserstein distance, which for $p = 2$ can be summarized as:

Theorem 4 (Bubenik [5])**.** *For any two persistence diagrams F and G we have*

$$
\begin{aligned}
&\|\Phi^L(F) - \Phi^L(G)\|_{L_2(\mathbb{R}^2)} \leq \\
&\inf_\gamma \left(\sum_{u \in F} \mathrm{p}(u) \|u - \gamma(u)\|_\infty^2 + \frac{2}{3} \|u - \gamma(u)\|_\infty^3 \right)^{\frac{1}{2}},
\end{aligned} \tag{12}
$$

where $\mathrm{p}(u) = d - b$ denotes the persistence of $u = (b,d)$, and γ ranges over all bijections from F to G.

For a better understanding of the stability results given in Theorems 2 and 4, we present and discuss two thought experiments.

For the first experiment, let $F_\lambda = \{-\lambda, \lambda\}$ and $G_\lambda = \{-\lambda + 1, \lambda + 1\}$ be two diagrams with one point each and $\lambda \in \mathbb{R}_{\geq 0}$. The two points move away from the diagonal with increasing λ, while maintaining the same Euclidean distance to each other. Consequently, $d_{W,p}(F_\lambda, G_\lambda)$ and $d_{k_\sigma}(F_\lambda, G_\lambda)$ asymptotically approach a constant as $\lambda \to \infty$. In contrast, $d_{k^L}(F_\lambda, G_\lambda)$ grows in the order of $\sqrt{\lambda}$ and, in particular, is unbounded. This means that d_{k^L} emphasizes points of high persistence in the diagrams, as reflected by the weighting term $\mathrm{p}(u)$ in (12).

In the second experiment, we compare persistence diagrams from data samples of two fictive classes A (*i.e.*, F,F') and B (*i.e.*, G), illustrated in Fig. 3. We first consider $d_{k^L}(F,F')$. As we have seen in the previous experiment, d_{k^L} will be dominated by variations in the points of high persistence. Similarly, $d_{k^L}(F,G)$ will also be dominated by these points as long as λ is sufficiently large. Hence, instances of classes A and B would be inseparable in a nearest neighbor setup. In contrast, d_B, $d_{W,p}$ and d_{k_σ} do *not* overemphasize points of high persistence and thus allow to distinguish classes A and B.

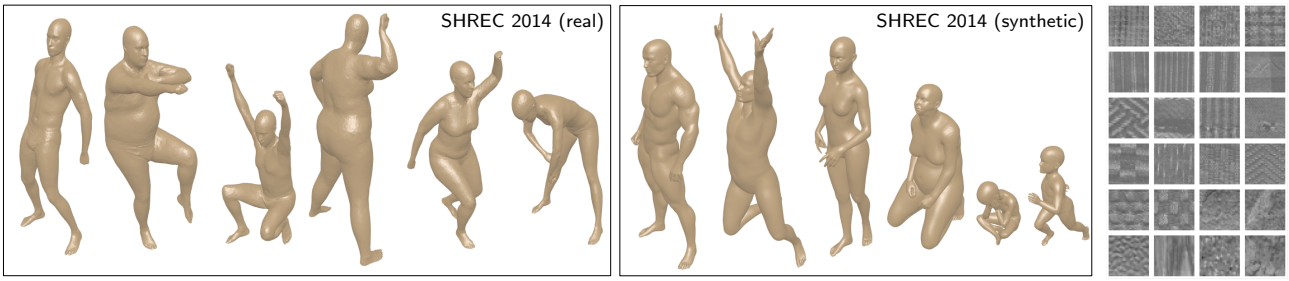

Figure 4: Examples from SHREC 2014 [24] (left, middle) and OuTeX `Outex_TC_00000` [22] (right).

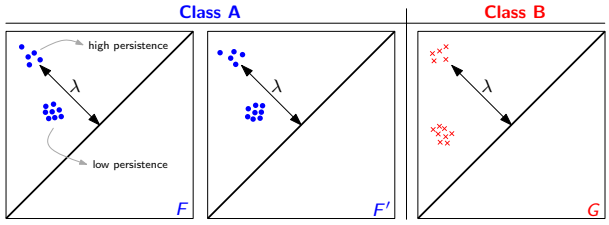

Figure 3: Two persistence diagrams F, F' from class A and one diagram G from class B. The classes only differ in their points of low-persistence (*i.e.*, points closer to the diagonal).

5.2. Empirical results

We report results on two vision tasks where persistent homology has already been shown to provide valuable discriminative information [20]: *shape classification/retrieval* and *texture image classification*. The purpose of the experiments is *not* to outperform the state-of-the-art on these problems – which would be rather challenging by exclusively using topological information – but to demonstrate the advantages of k_σ and d_{k_σ} over k^L and d_{k^L}.

Datasets. For shape classification/retrieval, we use the SHREC 2014 [24] benchmark, see Fig. 4. It consists of both *synthetic* and *real* shapes, given as 3D meshes. The synthetic part of the data contains 300 meshes of humans (five males, five females, five children) in 20 different poses; the real part contains 400 meshes from 40 humans (male, female) in 10 different poses. We use the meshes in full resolution, *i.e.*, without any mesh decimation. For classification, the objective is to distinguish between the different human models, *i.e.*, a 15-class problem for SHREC 2014 (synthetic) and a 40-class problem for SHREC 2014 (real).

For texture recognition, we use the `Outex_TC_00000` benchmark [22], downsampled to 32×32 pixel images. The benchmark provides 100 predefined training/testing splits and each of the 24 classes is equally represented by 10 images during training and testing.

Implementation. For shape classification/retrieval, we compute the classic *Heat Kernel Signature (HKS)* [29] over a range of ten time parameters t_i of increasing value. For each specific choice of t_i, we obtain a piecewise linear function on the surface mesh of each object. As discussed in Sec. 3, we then compute the persistence diagrams of the induced filtrations in dimensions 0 and 1.

For texture classification, we compute CLBP [16] descriptors, (*c.f.* [20]). Results are reported for the rotation-invariant versions of the CLBP-Single (`CLBP-S`) and the CLBP-Magnitude (`CLBP-M`) operator with $P = 8$ neighbours and radius $R = 1$. Both operators produce a scalar-valued response image which can be interpreted as a weighted cubical cell complex and its lower star filtration is used to compute persistence diagrams; see [30] for details.

For both types of input data, the persistence diagrams are obtained using Dipha [3], which can directly handle meshes and images. A standard soft margin C-SVM classifier [26], as implemented in Libsvm [7], is used for classification. The cost factor C is tuned using ten-fold cross-validation on the training data. For the kernel k_σ, this cross-validation further includes the kernel scale σ.

5.2.1 Shape classification

Tables 1 and 2 list the classification results for k_σ and k^L on SHREC 2014. All results are averaged over ten cross-validation runs using random 70/30 training/testing splits with a roughly equal class distribution. We report results for 1-dimensional features only; 0-dimensional features lead to comparable performance.

On both real and synthetic data, we observe that k_σ leads to consistent improvements over k^L. For some choices of t_i, the gains even range up to 30%, while in other cases, the improvements are relatively small. This can be explained by the fact that varying the HKS time t_i essentially varies the smoothness of the input data. The scale σ in k_σ allows to compensate—at the classification stage—for unfavorable smoothness settings to a certain extent, see Sec. 4. In contrast, k^L does not have this capability and essentially relies on suitably preprocessed input data. For some choices of t_i, k^L does in fact lead to classification accuracies close to k_σ. However, when using k^L, we have to carefully adjust the HKS time parameter, corresponding to changes in the in-

HKS t_i	k^L	k_σ	Δ
t_1	68.0 ± 3.2	94.7 ± 5.1	+26.7
t_2	**88.3** ± 3.3	**99.3** ± 0.9	+11.0
t_3	61.7 ± 3.1	96.3 ± 2.2	+34.7
t_4	81.0 ± 6.5	97.3 ± 1.9	+16.3
t_5	84.7 ± 1.8	96.3 ± 2.5	+11.7
t_6	70.0 ± 7.0	93.7 ± 3.2	+23.7
t_7	73.0 ± 9.5	88.0 ± 4.5	+15.0
t_8	81.0 ± 3.8	88.3 ± 6.0	+7.3
t_9	67.3 ± 7.4	88.0 ± 5.8	+20.7
t_{10}	55.3 ± 3.6	91.0 ± 4.0	+35.7

Table 1: Classification performance on SHREC 2014 (synthetic).

HKS t_i	k^L	k_σ	Δ
t_1	45.2 ± 5.8	48.8 ± 4.9	+3.5
t_2	31.0 ± 4.8	46.5 ± 5.3	+15.5
t_3	30.0 ± 7.3	37.8 ± 8.2	+7.8
t_4	41.2 ± 2.2	50.2 ± 5.4	+9.0
t_5	46.2 ± 5.8	62.5 ± 2.0	+16.2
t_6	33.2 ± 4.1	58.0 ± 4.0	+24.7
t_7	31.0 ± 5.7	**62.7** ± 4.6	+31.7
t_8	**51.7** ± 2.9	57.5 ± 4.2	+5.8
t_9	36.0 ± 5.3	41.2 ± 4.9	+5.2
t_{10}	2.8 ± 0.6	27.8 ± 5.8	+25.0

Table 2: Classification performance on SHREC 2014 (real).

put data. This is undesirable in most situations, since HKS computation for meshes with a large number of vertices can be quite time-consuming and sometimes we might not even have access to the meshes directly. The improved classification rates for k_σ indicate that using the additional degree of freedom is in fact beneficial for performance.

5.2.2 Shape retrieval

In addition to the classification experiments, we report on shape retrieval performance using standard evaluation measures (see [27, 24]). This allows us to assess the behavior of the kernel-induced distances d_{k_σ} and d_{k^L}.

For brevity, only the nearest-neighbor performance is listed in Table 3 (for a listing of all measures, see Appendix D). Using each shape as a query shape once, nearest-neighbor performance measures how often the top-ranked shape in the retrieval result belongs to the same class as the query. To study the effect of tuning the scale σ, the column d_{k_σ} lists the *maximum* nearest-neighbor performance that can be achieved over a range of scales.

As we can see, the results are similar to the classification experiment. However, at a few specific settings of the HKS time t_i, d_{k^L} performs on par, or better than d_{k_σ}. As noted in Sec. 5.2.1, this can be explained by the changes in the smoothness of the input data, induced by different HKS times t_i. Another observation is that nearest-neighbor performance of d_{k^L} is quite unstable around the top result with respect to t_i. For example, it drops at t_2 from 91% to 53.3% and 76.7% on SHREC 2014 (synthetic) and at t_8 from 70%

HKS t_i	d_{k^L}	d_{k_σ}	Δ	d_{k^L}	d_{k_σ}	Δ
t_1	53.3	88.7	+35.4	24.0	23.7	−0.3
t_2	**91.0**	**94.7**	+3.7	20.5	25.7	+5.2
t_3	76.7	91.3	+14.6	16.0	18.5	+2.5
t_4	84.3	93.0	+8.7	26.8	33.0	+6.2
t_5	85.0	92.3	+7.3	28.0	38.7	+10.7
t_6	63.0	77.3	+14.3	28.7	36.8	+8.1
t_7	65.0	80.0	+15.0	43.5	52.7	+9.2
t_8	73.3	80.7	+7.4	**70.0**	**58.2**	−11.8
t_9	73.0	83.0	+10.0	45.2	56.7	+11.5
t_{10}	51.3	69.3	+18.0	3.5	44.0	+40.5
Top-3 [24]	99.3 – 92.3 – 91.0			68.5 – 59.8 – 58.3		

Table 3: Nearest neighbor retrieval performance. *Left:* SHREC 2014 (synthetic); *Right:* SHREC 2014 (real).

to 45.2% and 43.5% on SHREC 2014 (real). In contrast, d_{k_σ} exhibits stable performance around the optimal t_i.

To put these results into context with existing works in shape retrieval, Table 3 also lists the top three entries (out of 22) of [24] on the same benchmark. On both real and synthetic data, d_{k_σ} ranks among the top five entries. This indicates that topological persistence alone is a rich source of discriminative information for this particular problem. In addition, since we only assess one HKS time parameter at a time, performance could potentially be improved by more elaborate fusion strategies.

5.3. Texture recognition

For texture recognition, all results are averaged over the 100 training/testing splits of the `Outex_TC_00000` benchmark. Table 4 lists the performance of a SVM classifier using k_σ and k^L for 0-dimensional features (*i.e.*, connected components). Higher-dimensional features were not informative for this problem. For comparison, Table 4 also lists the performance of a SVM, trained on normalized histograms of `CLBP-S/M` responses, using a χ^2 kernel.

First, from Table 4, it is evident that k_σ performs better than k^L by a large margin, with gains up to ≈11% in accuracy. Second, it is also apparent that, for this problem, topological information alone is not competitive with SVMs using simple orderless operator response histograms. However, the results of [20] show that a *combination* of persistence information (using persistence landscapes) with conventional bag-of-feature representations leads to state-of-the-art performance. While this indicates the complementary nature of topological features, it also suggests that kernel combinations (*e.g.*, via multiple-kernel learning [15]) could lead to even greater gains by including the proposed kernel k_σ.

To assess the stability of the (customary) cross-validation strategy to select a specific σ, Fig. 5 illustrates classification performance as a function of the latter. Given the smoothness of the performance curve, it seems unlikely that parameter selection via cross-validation will be sensitive to a

CLBP Operator	k^L	k_σ	Δ
CLBP-S	58.0 ± 2.3	**69.2 ± 2.7**	+11.2
CLBP-M	45.2 ± 2.5	**55.1 ± 2.5**	+9.9
CLBP-S (SVM-χ^2)	76.1 ± 2.2		
CLBP-M (SVM-χ^2)	76.7 ± 1.8		

Table 4: Classification performance on Outex_TC_00000.

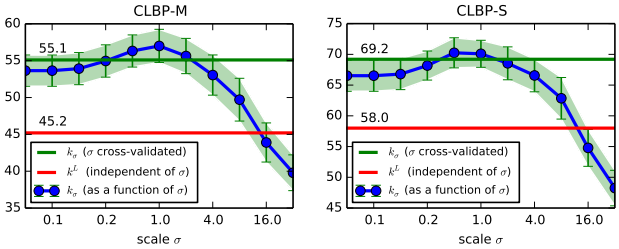

Figure 5: Texture classification performance of a SVM classifier with (1) the kernel k_σ as a function of σ, (2) the kernel k_σ with σ cross-validated and (3) the kernel k^L are shown.

specific discretization of the search range $[\sigma_{\min}, \sigma_{\max}]$.

Finally, we remark that tuning k^L has the same drawbacks in this case as in the shape classification experiments. While, in principle, we could smooth the textures, the CLBP response images, or even tweak the radius of the CLBP operators, all those strategies would require changes at the beginning of the processing pipeline. In contrast, adjusting the scale σ in k_σ is done at the *end* of the pipeline during classifier training.

6. Conclusion

We have shown, both theoretically and empirically, that the proposed kernel exhibits good behavior for tasks like shape classification or texture recognition using a SVM. Moreover, the ability to tune a scale parameter has proven beneficial in practice.

One possible direction for future work would be to address computational bottlenecks in order to enable application in large scale scenarios. This could include leveraging additivity and stability in order to approximate the value of the kernel within given error bounds, in particular, by reducing the number of distinct points in the summation of (10).

While the 1-Wasserstein distance is well established and has proven useful in applications, we hope to improve the understanding of stability for persistence diagrams w.r.t. the Wasserstein distance beyond the previous estimates. Such a result would extend the stability of our kernel from persistence diagrams to the underlying data, leading to a full stability proof for topological machine learning.

In summary, our method enables the use of topological information in all kernel-based machine learning methods. It will therefore be interesting to see which other application areas will profit from topological machine learning.

References

[1] A. Adcock, E. Carlsson, and G. Carlsson. The Ring of Algebraic Functions on Persistence Bar Codes. arXiv, available at http://arxiv.org/abs/1304.0530, 2013.

[2] R. Bapat and T. Raghavan. *Nonnegative Matrices and Applications*. Cambridge University Press, 1997.

[3] U. Bauer, M. Kerber, and J. Reininghaus. Distributed computation of persistent homology. In *ALENEX*, 2014.

[4] C. Berg, J.-P. Reus-Christensen, and P. Ressel. *Harmonic Analysis on Semi-Groups – Theory of Positive Definite and Related Functions*. Springer, 1984.

[5] P. Bubenik. Statistical topological data analysis using persistence landscapes. arXiv, available at http://arxiv.org/abs/1207.6437, 2012.

[6] G. Carlsson. Topology and data. *Bull. Amer. Math. Soc.*, 46:255–308, 2009.

[7] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2(3):1–27, 2011.

[8] F. Chazal, L. Guibas, S. Oudot, and P. Skraba. Persistence-based clustering in Riemannian manifolds. In *SoSG*, 2011.

[9] C. Chen, D. Freedman, and C. Lampert. Enforcing topological constraints in random field image segmentation. In *CVPR*, 2013.

[10] M. Chung, P. Bubenik, and P. Kim. Persistence diagrams of cortical surface data. In *IPMI*, 2009.

[11] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete Comp. Geom.*, 37(1):103–120, 2007.

[12] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz functions have L_p-stable persistence. *Found. Comput. Math.*, 10(2):127–139, 2010.

[13] H. Edelsbrunner and J. Harer. *Computational Topology. An Introduction.* AMS, 2010.

[14] M. Gao, C. Chen, S. Zhang, Z. Qian, D. Metaxas, and L. Axel. Segmenting the papillary muscles and the trabeculae from high resolution cardiac CT through restoration of topological handles. In *IPMI*, 2013.

[15] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *J. Mach. Learn. Res.*, 12:2211–2268, 2011.

[16] Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE TIP*, 19(6):16571663, 2010.

[17] T. Iijima. Basic theory on normalization of a pattern (in case of typical one-dimensional pattern). *Bulletin of Electrical Laboratory*, 26:368–388, 1962.

[18] R. J. j. Iorio and V. de Magalhães Iorio. *Fourier analysis and partial differential equations.* Cambridge Stud. Adv. Math., 2001.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[20] C. Li, M. Ovsjanikov, and F. Chazal. Persistence-based structural recognition. In *CVPR*, 2014.

[21] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[22] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllonen, and S. Huovinen. OuTeX – new framework for empirical evaluation of texture analysis algorithms. In *ICPR*, 2002.

[23] D. Pachauri, C. Hinrichs, M. Chung, S. Johnson, and V. Singh. Topology-based kernels with application to inference problems in Alzheimers disease. *IEEE TMI*, 30(10):1760–1770, 2011.

[24] Pickup, D. *et al.*. SHREC '14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, EG 3DOR'14. Eurographics Association, 2014.

[25] B. Schölkopf. The kernel-trick for distances. In *NIPS*, 2001.

[26] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[27] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling International*, 2004.

[28] P. Skraba, M. Ovsjanikov, F. Chazal, and L. Guibas. Persistence-based segmentation of deformable shapes. In *CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*, 2010.

[29] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and probably informative multi-scale signature based on heat diffusion. In *SGP*, 2009.

[30] H. Wagner, C. Chen, and E. Vuçini. Efficient computation of persistent homology for cubical data. In *Topological Methods in Data Analysis and Visualization II*, Mathematics and Visualization, pages 91–106. Springer Berlin Heidelberg, 2012.

Appendix

A. Indefiniteness of $d_{W,p}$

It is tempting to try to employ the Wasserstein distance for constructing a kernel on persistence diagrams. For instance, in Euclidean space, $k(x,y) = -\|x-y\|^2, x, y \in \mathbb{R}^n$ is conditionally positive definite and can be used within SVMs. Hence, the question arises if $k(x,y) = -d_{W,p}(x,y), x, y \in \mathcal{D}$ can be used as well.

In the following, we demonstrate (via counterexamples) that neither $-d_{W,p}$ nor $\exp(-\xi d_{W,p}(\cdot,\cdot))$ – for different choices of p – are (conditionally) positive definite. Thus, they cannot be employed in kernel-based learning techniques.

First, we briefly repeat some definitions to establish the terminology; this is done to avoid potential confusion, w.r.t. references [4, 2, 26]), about what is referred to as (conditionally) positive/negative definiteness in the context of kernel functions.

Definition 2. *A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is called positive definite (p.d.) if $\mathbf{c}^\top \mathbf{A}\mathbf{c} \geq 0$ for all $\mathbf{c} \in \mathbb{R}^n$. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is called negative definite (n.d.) if $\mathbf{c}^\top \mathbf{A}\mathbf{c} \leq 0$ for all $\mathbf{c} \in \mathbb{R}^n$.*

Note that in literature on linear algebra the notion of definiteness as introduced above is typically known as semidefiniteness. For the sake of brevity, in the kernel literature the prefix "semi" is typically dropped.

Definition 3. *A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is called conditionally positive definite (c.p.d.) if $\mathbf{c}^t\mathbf{A}\mathbf{c} \geq 0$ for all $\mathbf{c} = (c_1,\dots,c_n) \in \mathbb{R}^n$ s.t. $\sum_i c_i = 0$. A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is called conditionally negative definite (c.n.d.) if $\mathbf{c}^\top \mathbf{A}c \leq 0$ for all $\mathbf{c} = (c_1,\dots,c_n) \in \mathbb{R}^n$ s.t. $\sum_i c_i = 0$.*

Definition 4. *Given a set $\mathcal{X}$, a function $k\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is a* positive definite kernel *if there exists a Hilbert space $\mathcal{H}$ and a map $\Phi\colon \mathcal{X} \to \mathcal{H}$ such that $k(x,y) = \langle \Phi(x), \Phi(y)\rangle_{\mathcal{H}}$.*

Typically a positive definite kernel is simply called *kernel*. Roughly speaking, the utility of p.d. kernels comes from the fact that they enable the "kernel-trick", *i.e.*, the use of algorithms that can be formulated in terms of dot products in an implicit feature space [26]. However, as shown by Schölkopf in [25], this "kernel-trick" also works for distances, leading to the larger class of c.p.d. kernels (see Definition 5), which can be used in kernel-based algorithms that are translation-invariant (*e.g.*, SVMs or kernel PCA).

Definition 5. *A function $k\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is (conditionally) positive (negative, resp.) definite kernel if and only if k is symmetric and for every finite subset $\{x_1,\dots,x_m\} \subseteq \mathcal{X}$ the Gram matrix $(k(x_i,x_j))_{i,j=1,1}^{m,m}$ is (conditionally) positive (negative, resp.) definite.*

To demonstrate that a function is not c.p.d. or c.n.d., resp., we can look at the eigenvalues of the corresponding Gram matrices. In fact, it is known that a matrix $\mathbf{A}$ is p.d. if and only if all its eigenvalues are nonnegative. The following lemmas from [2] give similar, but weaker results for (nonnegative) c.n.d. matrices, which will be useful to us.

Lemma 5 (see Lemma 4.1.4 of [2])**.** *If $\mathbf{A}$ is a c.n.d. matrix, then $\mathbf{A}$ has at most one positive eigenvalue.*

Corollary 1 (see Corollary 4.1.5 of [2])**.** *Let $\mathbf{A}$ be a nonnegative, nonzero matrix that is c.n.d. Then $\mathbf{A}$ has exactly one positive eigenvalue.*

The following theorem establishes a relation between c.n.d. and p.d. kernels.

Theorem 6 (see Chapter 2, §2, Theorem 2.2 of [4])**.** *Let $\mathcal{X}$ be a nonempty set and let $k : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ be symmetric. Then k is a conditionally negative definite kernel if and only if $\exp(-\xi k(\cdot,\cdot))$ is a positive definite kernel for all $\xi > 0$.*

In the code (`test_negative_type_simple.m`)[3], we generate simple examples for which the Gram matrix $\mathbf{A} = (d_{W,p}(x_i,x_j))_{i,j=1,1}^{m,m}$ – for various choices of p – has at least two positive and two negative eigenvalue. Thus, it is neither (c.)n.d. nor (c.)p.d. according to Corollary 1. Consequently, the function $\exp(-d_{W,p})$ is not p.d. either, by virtue of Theorem 6. To run the MATLAB code, simply execute:

```
load options_cvpr15.mat;
test_negative_type_simple(options);
```

This will generate a short summary of the eigenvalue computations for a selection of values for p, including $p = \infty$ (bottleneck distance).

Remark. While our simple counterexamples suggest that typical kernel constructions using $d_{W,p}$ for different p (including $p = \infty$) do not lead to (c.)p.d. kernels, a formal assessment of this question remains an open research question.

B. Plots of the feature map Φ_σ

Given a persistence diagram D, we consider the solution $u\colon \Omega\times\mathbb{R}_{\geq 0} \to \mathbb{R}, (x,t) \mapsto u(x,t)$ of the following partial differential equation

$$\begin{aligned} \Delta_x u &= \partial_t u && \text{in } \Omega\times\mathbb{R}_{>0}, \\ u &= 0 && \text{on } \partial\Omega\times\mathbb{R}_{\geq 0}, \\ u &= \sum_{p\in D} \delta_p && \text{on } \Omega\times\{0\}. \end{aligned}$$

[3] https://gist.github.com/rkwitt/4c1e235d702718a492d3; the file `options_cvpr15.mat` can be found at: http://www.rkwitt.org/media/files/options_cvpr15.mat

To solve the partial differential equation, we extend the domain from Ω to $\mathbb{R}^2$ and consider for each $p \in D$ a Dirac delta δ_p and a Dirac delta $-\delta_{\overline{p}}$, as illustrated in Fig. 6 (left). By convolving $\sum_{p\in D} \delta_p - \delta_{\overline{p}}$ with a Gaussian kernel, see Fig. 6 (right), we obtain a solution $u\colon \mathbb{R}^2 \times \mathbb{R}_{\geq 0} \to \mathbb{R}, (x,t) \mapsto u(x,t)$ for the following partial differential equation:

$$\begin{aligned} \Delta_x u &= \partial_t u && \text{in } \mathbb{R}^2 \times \mathbb{R}_{>0}, \\ u &= \sum_{p\in D} \delta_p - \delta_{\overline{p}} && \text{on } \mathbb{R}^2 \times \{0\}. \end{aligned}$$

Restricting the solution u to $\Omega \times \mathbb{R}_{\geq 0}$, we then obtain the following solution $u\colon \Omega \times \mathbb{R}_{\geq 0} \to \mathbb{R}$,

$$u(x,t) = \frac{1}{4\pi t} \sum_{p\in D} e^{-\frac{\|x-p\|^2}{4t}} - e^{-\frac{\|x-\overline{p}\|^2}{4t}} \tag{13}$$

for the original partial differential equation and $t > 0$. This yields the feature map $\Phi_\sigma\colon \mathcal{D} \to L_2(\Omega)$:

$$\Phi_\sigma(D)\colon \Omega \to \mathbb{R}, \quad x \mapsto \frac{1}{4\pi\sigma} \sum_{p\in D} e^{-\frac{\|x-p\|^2}{4\sigma}} - e^{-\frac{\|x-\overline{p}\|^2}{4\sigma}}. \tag{14}$$

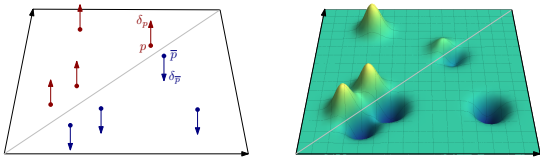

Figure 6: Solving the partial differential equation: First (left), we extend the domain from Ω to $\mathbb{R}^2$ and consider for each $p \in D$ a Dirac delta δ_p (red) and a Dirac delta $-\delta_{\overline{p}}$ (blue). Next (right), we convolve $\sum_{p\in D} \delta_p - \delta_{\overline{p}}$ with a Gaussian kernel.

In Fig. 7, we illustrate the effect of an increasing scale σ on the feature map $\Phi_\sigma(D)$. Note that in the right plot the influence of the low-persistence point close to the diagonal basically vanishes. This effect is essentially due to the Dirichlet boundary condition and is responsible for gaining stability for our persistence scale-space kernel k_σ.

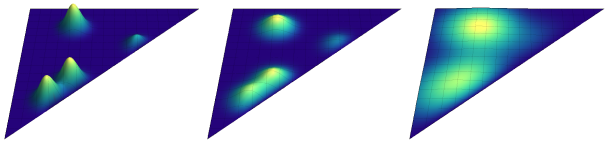

Figure 7: An illustration of the feature map $\Phi_\sigma(D)$ as a function in $L_2(\Omega)$ at growing scales σ (from left to right).

C. Closed-form solution for k_σ

For two persistence diagrams F and G, the persistence scale-space kernel $k_\sigma(F,G)$ is defined as $\langle \Phi_\sigma(F), \Phi_\sigma(G) \rangle_{L_2(\Omega)}$, which is

$$k_\sigma(F,G) = \int_\Omega \Phi_\sigma(F)\, \Phi_\sigma(G)\, dx.$$

By extending its domain from Ω to $\mathbb{R}^2$, we see that $\Phi_\sigma(D)(x) = -\Phi_\sigma(D)(\overline{x})$ for all $x \in \mathbb{R}^2$. Hence, $\Phi_\sigma(F)(x) \cdot \Phi_\sigma(G)(x) = \Phi_\sigma(F)(\overline{x}) \cdot \Phi_\sigma(G)(\overline{x})$ for all $x \in \mathbb{R}^2$, and we obtain

$$\begin{aligned} k_\sigma(F,G) &= \frac{1}{2} \int_{\mathbb{R}^2} \Phi_\sigma(F)\, \Phi_\sigma(G)\, dx \\ &= \frac{1}{2} \frac{1}{(4\pi\sigma)^2} \int_{\mathbb{R}^2} \left(\sum_{p\in F} e^{-\frac{\|x-p\|^2}{4\sigma}} - e^{-\frac{\|x-\overline{p}\|^2}{4\sigma}} \right) \cdot \\ &\quad \left(\sum_{q\in G} e^{-\frac{\|x-q\|^2}{4\sigma}} - e^{-\frac{\|x-\overline{q}\|^2}{4\sigma}} \right) dx \\ &= \frac{1}{2} \frac{1}{(4\pi\sigma)^2} \sum_{\substack{p\in F \\ q\in G}} \int_{\mathbb{R}^2} \left(e^{-\frac{\|x-p\|^2}{4\sigma}} - e^{-\frac{\|x-\overline{p}\|^2}{4\sigma}} \right) \cdot \\ &\quad \left(e^{-\frac{\|x-q\|^2}{4\sigma}} - e^{-\frac{\|x-\overline{q}\|^2}{4\sigma}} \right) dx \\ &= \frac{1}{(4\pi\sigma)^2} \sum_{\substack{p\in F \\ q\in G}} \int_{\mathbb{R}^2} e^{-\frac{\|x-p\|^2+\|x-q\|^2}{4\sigma}} - e^{-\frac{\|x-p\|^2+\|x-\overline{q}\|^2}{4\sigma}}\, dx. \end{aligned}$$

We calculate the integrals as follows:

$$\begin{aligned} \int_{\mathbb{R}^2} e^{-\frac{\|x-p\|^2+\|x-q\|^2}{4\sigma}}\, dx &= \int_{\mathbb{R}^2} e^{-\frac{\|x-(p-q)\|^2+\|x\|^2}{4\sigma}}\, dx \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} e^{-\frac{(x_1-\|p-q\|)^2+x_2^2+x_1^2+x_2^2}{4\sigma}}\, dx_1\, dx_2 \\ &= \int_{\mathbb{R}} e^{-\frac{x_2^2}{2\sigma}}\, dx_2 \cdot \int_{\mathbb{R}} e^{-\frac{(x_1-\|p-q\|)^2+x_1^2}{4\sigma}}\, dx_1 \\ &= \sqrt{2\pi\sigma} \cdot \int_{\mathbb{R}} e^{-\frac{(x_1-\|p-q\|)^2+x_1^2}{4\sigma}}\, dx_1 \\ &= \sqrt{2\pi\sigma} \cdot \int_{\mathbb{R}} e^{-\frac{(2x_1-\|p-q\|)^2+\|p-q\|^2}{8\sigma}}\, dx_1 \\ &= \sqrt{2\pi\sigma}\, e^{-\frac{\|p-q\|^2}{8\sigma}} \cdot \int_{\mathbb{R}} e^{-\frac{(2x_1-\|p-q\|)^2}{8\sigma}}\, dx_1 \\ &= \sqrt{2\pi\sigma}\, e^{-\frac{\|p-q\|^2}{8\sigma}} \cdot \int_{\mathbb{R}} e^{-\frac{x_1^2}{2\sigma}}\, dx_1 \\ &= 2\pi\sigma\, e^{-\frac{\|p-q\|^2}{8\sigma}}. \end{aligned}$$

In the first step, we applied a coordinate transform that moves $x - q$ to x. In the second step, we performed a rotation such that $p - q$ lands on the positive x_1-axis at distance $\|p - q\|$ to the origin and we applied Fubini's theorem. We finally obtain the closed-form expression for the kernel k_σ

as:

$$k_\sigma(F,G) = \frac{1}{(4\pi\sigma)^2} 2\pi\sigma \sum_{\substack{p\in F\\ q\in G}} e^{-\frac{\|p-q\|^2}{8\sigma}} - e^{-\frac{\|p-\bar{q}\|^2}{8\sigma}}$$
$$= \frac{1}{8\pi\sigma} \sum_{\substack{p\in F\\ q\in G}} e^{-\frac{\|p-q\|^2}{8\sigma}} - e^{-\frac{\|p-\bar{q}\|^2}{8\sigma}}.$$

D. Additional retrieval results on SHREC 2014

HKS t_i	d_{kL}	$d_{k\sigma}$	Δ	d_{kL}	$d_{k\sigma}$	Δ
t_1	59.9	71.3	+11.4	26.0	21.4	−4.6
t_2	**75.1**	76.0	+0.9	23.8	22.7	−1.1
t_3	49.6	64.8	+15.2	19.1	20.7	+1.6
t_4	59.4	**77.5**	+18.1	23.5	26.1	+2.6
t_5	68.1	75.2	+7.1	22.7	27.4	+4.7
t_6	50.0	55.2	+5.2	18.9	26.2	+7.3
t_7	47.6	53.6	+6.0	27.4	31.8	+4.4
t_8	53.1	62.4	+9.3	**45.3**	**39.8**	−5.5
t_9	51.2	56.3	+5.1	24.4	30.3	+5.9
t_{10}	39.6	49.7	+10.1	2.5	21.8	+19.3
Top-3 [24]	83.2 – 76.4 – 76.0			54.1 – 47.2 – 45.1		

Table 5: T1 retrieval performance. *Left:* SHREC 2014 (synthetic); *Right:* SHREC 2014 (real).

HKS t_i	d_{kL}	$d_{k\sigma}$	Δ	d_{kL}	$d_{k\sigma}$	Δ
t_1	87.7	91.4	+3.7	41.5	34.6	−6.9
t_2	**91.1**	**95.1**	+4.0	40.8	37.1	−3.7
t_3	70.4	83.4	+13.0	36.5	36.8	+0.3
t_4	77.7	93.6	+15.9	39.8	43.4	+3.6
t_5	90.8	92.3	+1.5	35.1	41.8	+6.7
t_6	73.9	75.4	+1.5	31.6	40.2	+8.6
t_7	70.6	74.4	+3.8	38.6	47.6	+9.0
t_8	73.3	79.3	+6.0	**56.5**	**57.6**	+1.1
t_9	72.7	76.2	+3.5	31.8	42.5	+10.7
t_{10}	57.8	66.6	+8.8	4.8	31.0	+26.2
Top-3 [24]	98.7 – 97.1 – 94.9			74.2 – 65.9 – 65.7		

Table 6: T2 retrieval performance. *Left:* SHREC 2014 (synthetic); *Right:* SHREC 2014 (real).

HKS t_i	d_{kL}	$d_{k\sigma}$	Δ	d_{kL}	$d_{k\sigma}$	Δ
t_1	60.6	65.3	+4.7	25.4	22.8	−2.6
t_2	**65.0**	67.4	+2.4	25.0	23.4	−1.6
t_3	48.4	58.8	+10.4	24.0	24.0	+0.0
t_4	55.2	**67.6**	+12.4	25.3	27.4	+2.1
t_5	63.7	66.2	+2.5	21.6	25.2	+3.6
t_6	51.0	52.7	+1.7	20.7	23.7	+3.0
t_7	48.4	51.7	+3.3	22.5	27.5	+5.0
t_8	51.1	56.5	+5.4	**30.2**	**33.2**	+3.0
t_9	50.4	53.2	+2.8	15.8	25.3	+9.5
t_{10}	39.8	46.7	+6.9	3.6	19.0	+15.4
Top-3 [24]	70.6 – 69.1 – 65.9			38.7 – 35.6 – 35.4		

Table 7: EM retrieval performance. *Left:* SHREC 2014 (synthetic); *Right:* SHREC 2014 (real).

HKS t_i	d_{kL}	$d_{k\sigma}$	Δ	d_{kL}	$d_{k\sigma}$	Δ
t_1	81.3	91.5	+10.2	53.0	49.6	−3.4
t_2	**92.1**	93.4	+1.3	51.1	51.3	+0.2
t_3	80.3	89.3	+9.0	47.7	48.4	+0.7
t_4	85.0	**93.8**	+8.8	52.7	55.5	+2.8
t_5	89.0	93.2	+4.2	51.2	55.5	+4.3
t_6	78.6	82.5	+3.9	48.1	54.2	+6.1
t_7	77.2	81.6	+4.4	55.7	60.5	+4.8
t_8	80.4	86.3	+5.9	**72.8**	**68.3**	−4.5
t_9	79.7	83.9	+4.2	50.4	61.0	+10.6
t_{10}	70.8	78.9	+8.1	27.7	51.3	+23.6
Top-3 [24]	97.7 – 93.8 – 92.7			78.1 – 71.7 – 71.2		

Table 8: DCG retrieval performance. *Left:* SHREC 2014 (synthetic); *Right:* SHREC 2014 (real).