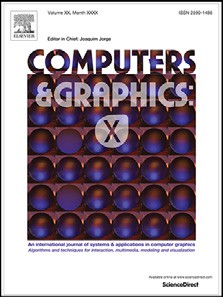
[Computers & Graphics: X 2 (2019) 100005](https://doi.org/10.1016/j.cagx.2019.100005)

Contents lists available at [ScienceDirect](http://www.ScienceDirect.com/)

Computers & Graphics: X

journal homepage: [www.elsevier.com/locate/cagx](http://www.elsevier.com/locate/cagx)

## Special Section on SMI 2019

A kernel for multi-parameter persistent homology

René Corbet [a,](#_bookmark0) Ulderico Fugacci[a,](#_bookmark0) Michael Kerber [a,](#_bookmark0) Claudia Landi[b,](#_bookmark1) Bei Wang [c,](#_bookmark2)[∗](#_bookmark3)

a *Graz University of Technology, Austria*

b *University of Modena and Reggio Emilia, Italy*

c *University of Utah, USA*

a r t i c l e i n f o a b s t r a c t

*Article history:*

Received 13 March 2019

Revised 18 May 2019

Accepted 21 May 2019

Available online 31 May 2019

*Keywords:*

Topological data analysis Machine learning Persistent homology Multivariate analysis

Topological data analysis and its main method, persistent homology, provide a toolkit for computing topological information of high-dimensional and noisy data sets. Kernels for one-parameter persistent homology have been established to connect persistent homology with machine learning techniques with applicability on shape analysis, recognition and classification. We contribute a kernel construction for multi-parameter persistence by integrating a one-parameter kernel weighted along straight lines. We prove that our kernel is stable and eﬃciently computable, which establishes a theoretical connection between topological data analysis and machine learning for multivariate data analysis.

© 2019 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license. (<http://creativecommons.org/licenses/by/4.0/>)

##### Introduction

Topological data analysis (TDA) is an active area in data sci- ence with a growing interest and notable successes in a number of applications in science and engineering [[1–8]](#_bookmark32). TDA extracts in- depth geometric information in amorphous solids [[5]](#_bookmark34), determines robust topological properties of evolution from genomic data sets

[[2]](#_bookmark33) and identifies distinct diabetes subgroups [[6]](#_bookmark35) and a new sub- type of breast cancer [[9]](#_bookmark36) in high-dimensional clinical data sets, to name a few. In the context of shape analysis, TDA techniques have been used in the recognition, classification [[10,11]](#_bookmark37), summarization [[12]](#_bookmark40), and clustering [[13]](#_bookmark41) of 2D/3D shapes and surfaces. Oftentimes, such techniques capture and highlight structures in data that con- ventional techniques fail to treat [[11,13]](#_bookmark38) or reveal properly [[5]](#_bookmark34).

TDA employs the mathematical notion of *simplicial complexes*

[[14]](#_bookmark42) to encode higher order interactions in the system, and at its [core uses the computational framework of *persistent homology* [15– 19] to extract multi-scale topological features of the data. In par-](#_bookmark43) ticular, TDA extracts a rich set of topological features from high- dimensional and noisy data sets that complement geometric and statistical features, which offers a different perspective for machine learning. The question is, *how can we establish and enrich the theo- retical connections between TDA and machine learning*?

Informally, *homology* was developed to classify topological spaces by examining their topological features such as con- nected components, tunnels, voids and holes of higher dimensions;

∗ Corresponding author.

[*E-mail addresses:* maths@rene-corbet.de (R. Corbet), kerber@tugraz.at (M. Kerber),](mailto:kerber@tugraz.at) [clandi@unimore.it](mailto:clandi@unimore.it) [(C. Landi),](mailto:kerber@tugraz.at) [beiwang@sci.utah.edu](mailto:beiwang@sci.utah.edu) [(B. Wang](mailto:kerber@tugraz.at)).

*persistent homology* studies homology of a data set at multiple scales. Such information is summarized by the *persistence diagram*, a finite multi-set of points in the plane. A persistence diagram yields a complete description of the topological properties of a data set, making it an attractive tool to define features of data that take topology into consideration. Furthermore, a celebrated theorem of persistent homology is the *stability* of persistence diagrams [[20]](#_bookmark45) – small changes in the data lead to small changes of the correspond- ing diagrams, making it suitable for robust data analysis.

However, interfacing persistence diagrams directly with ma- chine learning poses technical diﬃculties, because persistence diagrams contain point sets in the plane that do not have the structure of an inner product, which allows length and angle to be measured. In other words, such diagrams lack a Hilbert space structure for kernel-based learning methods such as kernel SVMs or PCAs [[21]](#_bookmark47). Recent work proposes several variants of *feature maps* [[21–23]](#_bookmark47) that transform persistence diagrams into *L*2-functions over R2. This idea immediately enables the application of topological features for kernel-based machine learning methods as establish- ing a kernel function implicitly defines a Hilbert space structure [[21]](#_bookmark47).

A serious limit of standard persistent homology and its initial interfacing with machine learning [[21–25]](#_bookmark47) is the restriction to only a single scale parameter, thereby confining its applicability to the univariate setting. However, in many real-world applica- tions, such as data acquisition and geometric modeling, we often encounter richer information described by multivariate data sets [[26–28]](#_bookmark53). Consider, for example, climate simulations where mul- tiple physical parameters such as temperature and pressure are computed simultaneously; and we are interested in understanding

<https://doi.org/10.1016/j.cagx.2019.100005>

2590-1486/© 2019 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license. (<http://creativecommons.org/licenses/by/4.0/>)

the interplay between these parameters. Consider another example in multivariate shape analysis, various families of functions carry information about the geometry of 3D shape objects, such as mesh density, eccentricity [[29]](#_bookmark58) or Heat Kernel Signature [[30]](#_bookmark39); and we are interested in creating multivariate signatures of shapes from such functions. Unlike the univariate setting, very few topological tools exist for the study of multivariate data [[29,31,32]](#_bookmark58), let alone the integration of multivariate topological features with machine learning.

The active area of *multi-parameter persistent homology*

[[26]](#_bookmark53) studies the extension of persistence to two or more (in- dependent) scale parameters. A complete discrete invariant such as the persistence diagram does not exist for more than one parameter [[26]](#_bookmark53). To gain partial information, it is common to study *slices*, that is, one-dimensional aﬃne subspaces where all parameters are connected by a linear equation. In this paper, we establish, for the first time, a theoretical connection between topo- logical features and machine learning algorithms via the kernel approach for multi-parameter persistent homology. Such a theo- retical underpinning is necessary for applications in multivariate data analysis.

*Our contribution.* We propose the first kernel construction for multi-parameter persistent homology. Our kernel is *generic*, *sta- ble* and can be *approximated in polynomial time*. For simplicity, we formulate all our results for the case of two parameters, although they extend to more than two parameters.

Our input is a data set that is filtered according to two scale pa- rameters and has a finite description size; we call this a *bi-filtration* and postpone its formal definition to [Section 2](#_bookmark4). Our main contribu- tion is the definition of a feature map that assigns to a bi-filtration

*X* a function *ФX* : *∆(*2*)* → R*,* where *∆*(2) is a subset of R4. More-

*Related work.* Our approach heavily relies on the construction of stable and eﬃciently computable feature maps for mono- filtrations. This line of research was started by Reininghaus et al. [[21]](#_bookmark47), whose approach we discuss in some detail in [Section 2](#_bookmark4). Alter- native kernel constructions appeared in [[24,33]](#_bookmark51). Kernel construc- tions fit into the general framework of including the space of persistence diagrams in a larger space with more favorable proper- ties. Other examples of this idea are persistent landscapes [[22]](#_bookmark49) and persistent images [[34]](#_bookmark43), which can be interpreted as kernel con- structions as well. Kernels and related variants defined on mono- filtrations have been used to discriminate and classify shapes and surfaces [[21,25]](#_bookmark47). An alternative approach comes from the defini- tion of suitable (polynomial) functions on persistence diagrams to arrive at a fixed-dimensional vector in R*d* on which machine learn- ing tasks can be performed; see [[35–38]](#_bookmark44).

As previously mentioned, a persistence diagram for multi- parameter persistence does not exist [[26]](#_bookmark53). However, bi-filtrations still admit meaningful distance measures, which lead to the notion of closeness of two bi-filtrations. The most prominent such dis- tance is the *interleaving distance* [[39]](#_bookmark46), which, however, has recently been proved to be NP-complete to compute and approximate [[40]](#_bookmark48). Computationally attractive alternatives are (multi-parameter) bot- tleneck distance [[41]](#_bookmark49) and the *matching distance* [[42,43]](#_bookmark50), which compares the persistence diagrams along all slices (appropriately weighted) and picks the worst discrepancy as the distance of the bi-filtrations. This distance can be approximated up to a precision

*‹* using an appropriate subsample of the lines [[42]](#_bookmark50), and also com-

puted exactly in polynomial time [[43]](#_bookmark51). Our approach extends these works in the sense that not just a distance, but an inner product on bi-filtrations, is defined with our inclusion into a Hilbert space. In a similar spirit, the software library RIVET [[44]](#_bookmark52) provides a visual- ization tool to explore bi-filtrations by scanning through the slices.

over, *Ф*2 is integrable over *∆*(2), effectively including the space of

*X*

bi-filtrations into the Hilbert space *L*2(*∆*(2)). Therefore, based on the standard scalar product in *L*2(*∆*(2)), a 2-parameter kernel is de- fined such that for two given bi-filtrations *X* and *Y* we have

*∆*

##### Preliminaries

We introduce the basic topological terminology needed in this

(*X , Y*⟩*Ф* :=

∫ *(*2*)*

*ФX ФY dμ.* (1)

work. We restrict ourselves to the case of simplicial complexes as input structures for a clearer geometric intuition of the con- cepts, but our results generalize to more abstract input types (such

We construct our feature map by interpreting a point of *∆*(2) as a

pair of (distinct) points in R2 that define a unique slice. Along this slice, the data simplifies to a *mono-filtration* (i.e., a filtration that depends on a single scale parameter), and we can choose among a large class of feature maps and kernel constructions of standard, one-parameter persistence. To make the feature map well-defined, we restrict our attention to a finite rectangle *R*.

Our inclusion into a Hilbert space induces a distance between bi-filtrations as

*dФ(X , Y)* := ,∫ *(ФX* − *ФY )*2 *dμ.* (2)

We prove a stability bound, relating this distance measure to the matching distance and the interleaving distance (see the paragraph on related work below). We also show that this stability bound is tight up to constant factors (see [Section 4](#_bookmark14)).

Finally, we prove that our kernel construction admits an eﬃ- cient approximation scheme. Fixing an absolute error bound *‹*, we give a polynomial time algorithm in 1/*‹* and the size of the bi- filtrations *X* and *Y* to compute a value *r* such that *r* ≤ (*X , Y*⟩*Ф* ≤ *r* + *‹*. On a high level, the algorithm subdivides the domain into

boxes of smaller and smaller width and evaluates the integral of

[(1)](#_bookmark5) by lower and upper sums within each subdomain, terminat- ing the process when the desired accuracy has been achieved. The technical diﬃculty lies in the accurate and certifiable approxima- tion of the variation of the feature map when moving the argu- ment within a subdomain.

as minimal representations of persistence modules) without prob- lems.

*Mono-filtrations.* Given a vertex set *V*, an *(abstract) simplex* is a non-empty subset of *V*, and an *(abstract) simplicial complex* is a col- lection of such subsets that is closed under the operation of taking non-empty subsets. A *subcomplex* of a simplicial complex *X* is a simplicial complex *Y* with *Y*⊆*X*. Fixing a finite simplicial complex *X*, a *mono-filtration X* of *X* is a map that assigns to each real num- ber *α*, a subcomplex *X (α)* of *X*, with the property that whenever *α* ≤ *β*, *X (α)* ⊆ *X (β)*. The *size* of *X* is the number of simplices of *X*. Since *X* is finite, *X (α)* changes at only finitely many places when *α* grows continuously from −∞ to +∞; we call these values *critical*. More formally, *α* is *critical* if there exists no open neighborhood of *α* such that the mono-filtration assigns the identical subcomplex to each value in the neighborhood. For a simplex *σ* of *X*, we call the *critical value* of *σ* the infimum over all *α* for which *σ* ∈ *X (α)*. For simplicity, we assume that this infimum is a minimum, so ev- ery simplex has a unique critical value wherever it is included in the mono-filtration.

*Bi-filtrations.* For points in R2*,* we write (*a*, *b*) ≤ (*c*, *d*) if *a* ≤ *c* and *b* ≤ *d*. Similarly, we say (*a*, *b*) *<* (*c*, *d*) if *a < c* and *b < d*. For a fi- nite simplicial complex *X*, a *bi-filtration X* of *X* is a map that as- signs to each point *p* ∈ R2 a subcomplex *X (p)* of *X*, such that whenever *p* ≤ *q*, *X (p)* ⊆ *X (q)*. Again, a point *p* = *(p*1*, p*2 *)* is called *critical* for *X* if, for any *‹ >* 0, both *X (p*1 − *‹, p*2 *)* and *X (p*1 *, p*2 − *‹)*

**Fig. 1.** The three black points mark the three critical points of some simplex *σ* in

*X*. The shaded area denotes the positions at which *σ* is present in the bi-filtration. Along the given slice (red line), the dashed lines denote the first position where the corresponding critical point “affects” the slice. This position is either the upper- vertical, or right-horizontal projection of the critical point onto the slice, depending on whether the critical point is below or above the line. For *σ* , we see that it enters the slice at the position marked by the blue point.

are not identical to *X (p)*. Note that unlike in the mono-filtration case, the set of critical points might not be finite. We call a bi- filtration *tame* if it has only finitely many such critical points. For a simplex *σ* , a point *p* ∈ R2 is *critical* for *σ* if, for any *‹ >* 0, *σ* is neither in *X (p*1 − *‹, p*2 *)* nor in *X (p*1 *, p*2 − *‹),* whereas *σ* is in both *X (p*1 + *‹, p*2 *)* and *X (p*1 *, p*2 + *‹)*. Again, for simplicity, we as- sume that *σ* ∈ *X (p)* in this case. A consequence of tameness is that each simplex has a finite number of critical points. Therefore, we can represent a tame bi-filtration of a finite simplicial com- plex *X* by specifying the set of critical points for each simplex in

*X*. The sum of the number of critical points over all simplices of *X* is called the *size* of the bi-filtration. We henceforth assume that bi-filtrations are always represented in this form; in particular, we assume tameness throughout this paper.

A standard example to generate bi-filtrations is by an arbitrary function *F* : *X* → R2 with the property that if *τ* ⊂ *σ* are two sim- plices of *X*, *F*(*τ* ) ≤ *F*(*σ* ). We define the *sublevel set X F (p)* as

*X F (p)* := {*σ* ∈ *X* | *F (σ )* ≤ *p*}*,*

and let *X F* denote its corresponding *sublevel set bi-filtration*. It is easy to verify that *X F* yields a (tame) bi-filtration and *F*(*σ* ) is the unique critical value of *σ* in the bi-filtration.

*Slices of a bi-filtration.* A bi-filtration *X* contains an infinite collec- tion of mono-filtrations. Let *L* be the set of all non-vertical lines in R2 with positive slope. Fixing any line *4* ∈ *L,* we observe that when traversing this line in positive direction, the subcomplexes of the bi-filtration are nested in each other. Note that *4* intersects the anti-diagonal *x* = −*y* in a unique base point *b*. Parameterizing

entire theory (which is explained, for instance, in [[45]](#_bookmark54)), we give an intuitive description here.

Persistent homology measures how the topological features of a data set evolve when considered across a varying scale parameter *α*. The most common example involves a point cloud in R*d,* where

considering a fixed scale *α* means replacing the points by balls of

radius *α*. As *α* increases, the data set undergoes various topological configurations, starting as a disconnected point cloud for *α* = 0 and ending up as a topological ball when *α* approaches ; see [Fig. 2](#_bookmark8)(a) for an example in R2.

∞

The topological information of this process can be summarized as a finite multi-set of points in the plane, called the *persistence diagram*. Each point of the diagram corresponds to a topological feature (i.e., connected components, tunnels, voids, etc.), and its coordinates specify at which scales the feature appears and dis- appears in the data. As illustrated in [Fig. 2](#_bookmark8)(a), all five (connected) components are born (i.e., appear) at *α* = 0. The green component dies (i.e., disappears) when it merges with the red component at *α* = 2*.*5; similarly, the orange, blue and pink components die at scales 3, 3.2 and 3.7, respectively. The red component never dies as *α* goes to ∞. The 0-dimensional persistence diagram is defined to have one point per component with birth and death value as its coordinates ([Fig. 2](#_bookmark8)(c)). The *persistence* of a feature is then merely its distance from the diagonal. While we focus on the components, the concept generalizes to higher dimensions, such as tunnels (1- dimensional homology) and voids (2-dimensional homology). For instance, in [Fig. 2](#_bookmark8)(a), a tunnel appears at *α* = 4*.*2 and disappears at *α* = 5*.*6*,* which gives rise to a purple point (4.2, 5.6) in the 1- dimensional persistence diagram ([Fig. 2](#_bookmark8)(c)).

From a computational point of view, the nested sequence of spaces formed by unions of balls ([Fig. 2](#_bookmark8)(a)) can be replaced by a nested sequence of simplicial complexes by taking their nerves, thereby forming a mono-filtration of simplicial complexes that cap- tures the same topological information but has a much smaller footprint ([Fig. 2](#_bookmark8)(b)).

In the context of shape analysis, we apply persistent homology to capture the topological information of 2D and 3D shape objects by employing various types of mono-filtrations. A simple example is illustrated in [Fig. 3](#_bookmark13): we extract point clouds sampled from the boundary of 2D shape objects and compute the persistence dia- grams using Vietoris-Rips complex filtrations.

*Stability of persistent homology.* Bottleneck distance represents a similarity measure between persistence diagrams. Let *D*, *D*r be two persistence diagrams. Without loss of generality, we can assume that both contain infinitely many copies of the points on the diag- onal. The *bottleneck distance* between *D* and *D*r is defined as

*dB (D, D*r *)* := inf sup  *x* − *γ (x) * ∞*,* (3)

*4* as *b* + *λ* · *a,* where *a* is the (positive) unit direction vector of *4*,

*γ x*∈*D*

we obtain the mono-filtration

*X4 (α)* := *X (b* + *α* · *a).*

We will refer to this mono-filtration *X4* as a *slice* of *X* along *4* (and sometimes also call *4* itself the slice, abusing notation). The crit- ical values of a slice can be inferred by the critical points of the bi-filtration in a computationally straightforward way. Instead of a formal description, we refer to [Fig. 1](#_bookmark6) for a graphical description. Also, if the bi-filtration is of size *n*, each of its slices is of size at most *n*.

*Persistent homology.* A mono-filtration *X* gives rise to a persistence diagram. Formally, we obtain this diagram by applying the homol- ogy functor to *X ,* yielding a sequence of vector spaces and linear maps between them, and splitting this sequence into indecompos- able parts using representation theory. Instead of rolling out the

where *γ* ranges over all bijections from *D* to *D*r. We will also

use the notation *dB (X , Y)* for two mono-filtrations instead of

*dB (D(X ), D(Y))*

A crucial result for persistent homology is the *stability theo- rem* proven in [[46]](#_bookmark55) and re-stated in our notation as follows. Given two functions *f, g* : *X* → R whose sublevel sets form two mono- filtrations of a finite simplicial complex *X*, the induced persistence diagrams satisfy

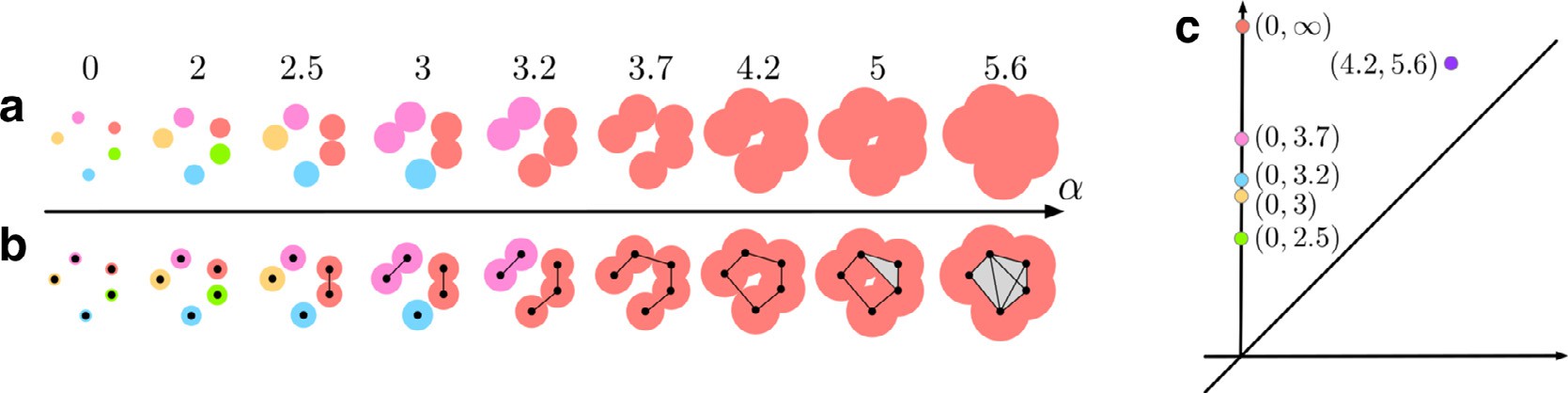
*dB (D f , Dg )* ≤  *f* − *g * ∞ := sup | *f (σ )* − *g(σ )*|*.* (4)

*σ* ∈*X*

*Feature maps for mono-filtrations.* Several feature maps aimed at the construction of a kernel for mono-filtrations have been pro- posed in the literature [[21–23]](#_bookmark47). We discuss one example: the persistence scale-space kernel [[21]](#_bookmark47) assigns to a mono-filtration

© }

*X* an *L*2-function *φX* defined on *∆(*1*)* := *(x*1 *, x*2 *)* ∈ R2 | *x*1 *< x*2 .



**Fig. 2.** Computing persistent homology of a point cloud in R2 . (a) A nested sequence of topological spaces formed by unions of balls at increasing parameter values. (b) A mono-filtration of simplicial complexes that captures the same topological information as in (a). (c) 0-dimensional and 1-dimensional persistence diagrams combined.

*4*

The main idea behind the definition of *φX* is to define a sum of Gaussian peaks, all of the same height and width, with each peak centered at one finite off-diagonal point of the persistence diagram *D(X )* of *X* . To make the construction robust against perturbations, the function has to be equal to 0 across the diagonal (the boundary of *∆*(1)). This is achieved by adding negative Gaussian peaks at the reflections of the off-diagonal points along the diagonal. Writing *z*¯ for the reflection of a point *z*, we obtain the formula,

Fixing (*p*, *q*) ∈ *∆*(2), let *4* denote the unique slice through these two points. Along this slice, the bi-filtration gives rise to a mono- filtration *X4,* and consequently a function *φX* : *∆(*1*)* → R using the considered feature map for mono-filtrations. Moreover, using the parameterization of the slice *4* as *b* + *λ* · *a* from [Section 2](#_bookmark4), there ex- ist real values *λp*, *λq* such that *b* + *λpa* = *p* and *b* + *λqa* = *q*. Since *p < q* and *λp < λq*, hence (*λp*, *λq*) ∈ *∆*(1). We define *ФX (p, q)* to be the weighted function value of *φX4* at (*λp*, *λq*) (see also [Fig. 4](#_bookmark15)), that

*φX (x)*

### 1

:= 4*πt*

*z*∈*D(X )*

Σ

** *x*−*z * 2

*e* 4*t* − *e*

2

** *x*−*z*¯  2

4*t ,* (5)

2

is,

*ФX (p, q)* := *w(p, q)* · *φX4 (λp, λq ),* (6)

where *t* is the width of the Gaussian, which is a free parameter of the construction. See [Fig. 4](#_bookmark15)(b) and (c) for an illustration of a transformation of a persistence diagram to the function *φX* . The induced kernel enjoys several stability properties and can be eval- uated eﬃciently without explicit construction of the feature map; see [[21]](#_bookmark47) for details.

More generally, in this paper, we look at the class of all feature maps that assign to a mono-filtration *X* a function in *L*2(*∆*(1)). For such a feature map *φX ,* we define the following properties:

* *Absolutely boundedness.* There exists a constant *v*1 *>* 0 such that, for any mono-filtration *X* of size *n* and any *x* ∈ *∆*(1), 0 ≤ *φX (x)* ≤ *v*1 · *n*.
* *Lipschitzianity.* There exists a constant *v*2 *>* 0 such that, for any mono-filtration *X* of size *n* and any *x*, *x*r ∈ *∆*(1), |*φX (x)* − *φX (x*r *)* ≤ *v*2 · *n* · *x* − *x*r 2 .

|

* *Internal stability.* There exists a constant *v*3 *>* 0 such that, for

any pair of mono-filtrations *X , Y* of size *n* and any *x* ∈ *∆*(1),

|*φX (x)* − *φY (x)*| ≤ *v*3 · *n* · *dB (X , Y)*.

* *Efficiency.* For any *x* ∈ *∆*(1), *φX (x)* can be computed in polyno-

mial time in the size of *X ,* that is, in *O*(*nk*) for some *k* ≥ 0.

where *w(p, q)* is a weight function *w* : *∆(*2*)* → R defined below.

The weight function *w* has two components. First, let *R* be a bounded axis-aligned rectangle in R2; its bottom-left corner coin- cides with the origin of the coordinate axes. We define *w* such that its weight is 0 if *p* or *q* is outside of *R*. Second, for pairs of points within *R* × *R*, we assign a weight depending on the slope of the in- duced slices. Formally, let *4* be parameterized as *b* + *λ* · *a* as above, and recall that *a* is a unit vector with non-negative coordinates.

Write *a* = *(a*1 *, a*2 *)* and set *4*ˆ := min{*a*1 *, a*2 }. Then, we define

*w(p, q)* := *χR (p)* · *χR (q)* · *4*ˆ*,*

where *χR* is the characteristic function of *R*, mapping a point *x* to 1 if *x* ∈ *R* and 0 otherwise.

The factor *4*ˆ ensures that slices that are close to being horizon-

tal or vertical attain less importance in the feature map. The same weight is assigned to slices in the matching distance [[42]](#_bookmark50). *4*ˆ is not important for obtaining an *L*2-function, but its meaning will be- come clear in the stability results of [Section 4](#_bookmark14). We also remark that the largest weight is attained for the diagonal slice with a value of

√

1*/* 2. Consequently, *w* is a non-negative function upper bounded

It can be verified easily that the scale-space feature map from above satisfies all these properties. The same is true, for instance, if the Gaussian peaks are replaced by linear peaks (that is, replacing the Gaussian kernel in [(5)](#_bookmark9) by a triangle kernel).

##### A feature map for multi-parameter persistent homology

Let *φ* be a feature map (such as the scale-space kernel) that assigns to a mono-filtration a function in *L*2(*∆*(1)). Starting from *φ*, we construct a feature map *Ф* on the set of all bi-filtrations *▲* that has values in a Hilbert space.

The feature map *Ф* assigns to a bi-filtration *X* a function *ФX* :

*∆(*2*)* → R. We set

by 1*/*√2.

To summarize, our map *Ф* depends on the choice of an axis-aligned rectangle *R* and a choice of feature map for mono- filtrations, which itself might have associated parameters. For in- stance, using the scale-space feature map requires the choice of the width *t* (see [(5)](#_bookmark9)). It is only left to argue that the image of the feature map *Ф* is indeed an *L*2-function.

**Theorem 1.** *If φ is absolutely bounded, then ФX is in L*2(*∆*(2))*.*

**Proof.** Let *X* be a bi-filtration of size *n*. As mentioned earlier, each slice *X4* is of a size at most *n*. By absolute boundedness and the fact that the weight function is upper bounded by √1 *,* it follows

*v*1*n*

*∆(*2*)* :=

©*(p, q)* | *p* ∈ R2*, q* ∈ R2*, p < q*}

2

that |*ФX (p, q)*| ≤ √2 for all (*p*, *q*). Since the support of *ФX* is com-

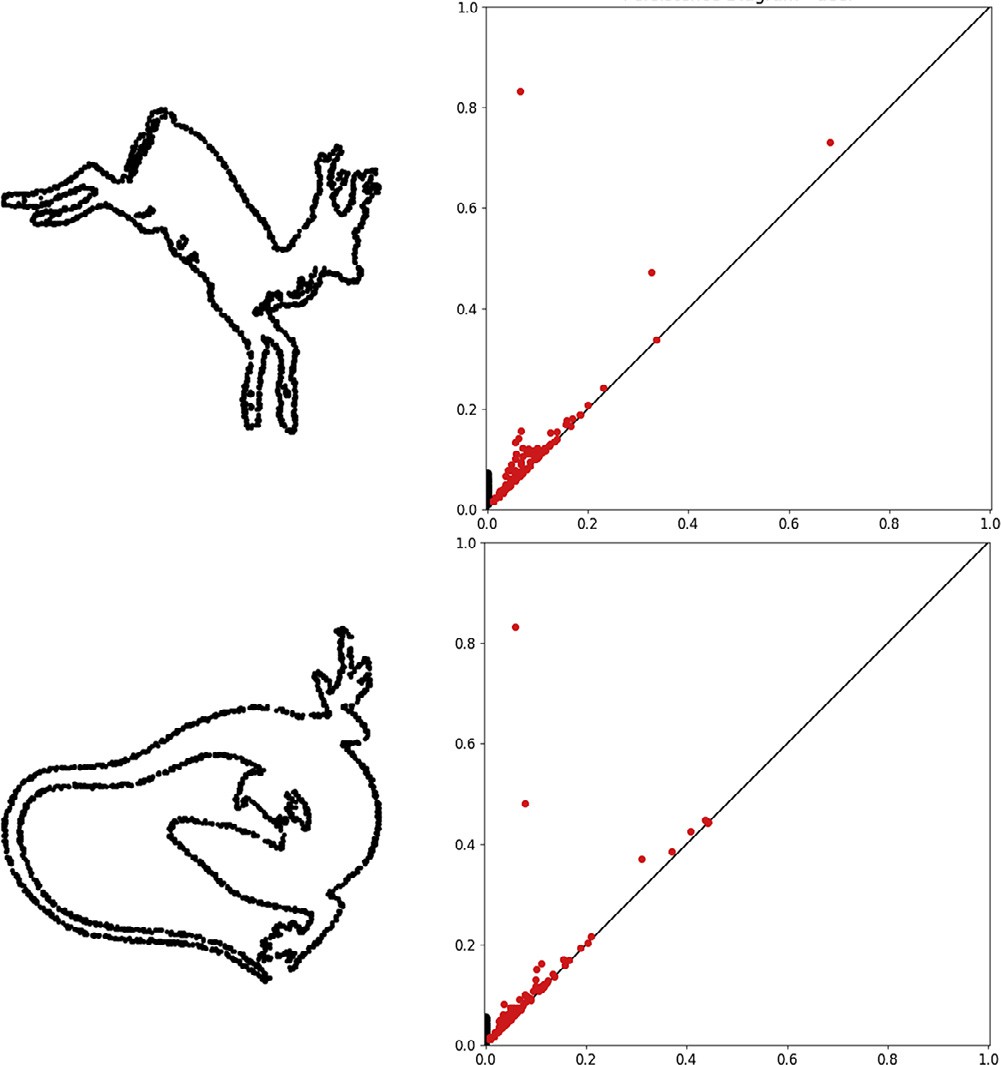
pact (*R* × *R*), the integral of *Ф*2 over *∆*(2) is finite, being absolutely

*X*

as the set of all pairs of points where the first point is smaller than the second one. *∆*(2) can be interpreted naturally as a subset of R4*,* but we will usually consider elements of *∆*(2) as pairs of points in R2.

bounded and compactly supported. ¤

Note that [Theorem 1](#_bookmark11) remains true even without restrict- ing the weight function to *R*, provided we consider a weight

bound

= *(v*3 · *n)*2 ∫

*∆(*2*)* ∩*(R*×*R)*

*(4*ˆ · *dB (X4, Y4 ))*2*dμ*

≤ *(v* · *n)*2 ∫ sup ¡*4*ˆ · *d (X , Y )*¢2 *dμ*

c ˛z I

3

*4*∈*L*

*B*

*4*

*4*

*∆* 2*)* ∩*(R*×*R)*

*(*

=*dmatch (X,Y)*

= *(v*3 · *n* · *d*

*match*

*(X , Y))*2 ∫

1*dμ.*

*∆(*2*)* ∩*(R*×*R)*

The claimed inequality follows by noting that the final integral is equal to 1 area*(R)*2. ¤

4

As a corollary, we get the the same stability statement with respect to interleaving distance instead of matching distance [[48](#_bookmark57), Thm.1]. Furthermore, we obtain a stability bound for sublevel set bi-filtrations of functions *X* → R2 [[47](#_bookmark56), Thm.4]:

**Corollary 3.** *Let F, G* : *X* → R2 *be two functions that give rise to sublevel set bi-filtrations X and Y, respectively. If φ is absolutely bounded and internally stable, we have*

 *ФX* − *ФY * *L*2 ≤ *C* · *n* · area*(R)* · *F* − *G * ∞*,*

*for some constant C.*

**Fig. 3.** The persistence diagrams of 2D shape objects. Black and red points are 0- dimensional and 1-dimensional features respectively (ignoring points with ∞ per- sistence).

function that is square-integrable over *∆*(2). We skip the (easy) proof.

##### Stability

An important and desirable property for a kernel is its stability. In general, stability means that small perturbations in the input data imply small perturbations in the output data. In our setting, small changes between multi-filtrations (with respect to matching distance) should not induce large changes in their corresponding feature maps (with respect to *L*2 distance).

Adopted to our notation, the matching distance is defined as

*dmatch(X , Y)* = sup ¡*4*ˆ · *dB (X4, Y4 )*¢*,*

*4*∈*L*

where *L* is the set of non-vertical lines with positive slope [[47]](#_bookmark56).

**Theorem 2.** *Let X and Y be two bi-filtrations. If φ is absolutely bounded and internally stable, we have*

 *ФX* − *ФY * *L*2 ≤ *C* · *n* · area*(R)* · *dmatch (X , Y),*

*for some constant C.*

We remark that the appearance of *n* in the stability bound is not desirable as the bound worsens when the complex size in- creases (unlike, for instance, the bottleneck stability bound in [(4)](#_bookmark7), which is independent of *n*). The factor of *n* comes from the inter- nal stability property of *φ*, so we have to strengthen this condition on *φ*. However, we show that such an improvement is impossible for a large class of “reasonable” feature maps.

For two bi-filtrations *X , Y* we define *X* *Y* by setting *(X* *Y)(p)* := *X (p)* п *Y(p)* for all *p* ∈ R2. A feature map *Ф* is *additive* if *ФX* ⊕*Y* = *Ф(X )* + *Ф(Y)* for all bi-filtrations *X , Y*. *Ф* is called *non- trivial* if there is a bi-filtration *X* such that *Ф L*2 /= 0. Additivity and non-triviality for feature maps *φ* on mono-filtrations is de-

fined in the analogous way. Note that, for instance, the scale space feature map is additive. Moreover, because *(X* *Y)4* = *X4* *Y4* for every slice *4*, a feature map *Ф* is additive if the underlying *φ* is additive.

For mono-filtrations, no additive, non-trivial feature map *φ* can satisfy

 *φX* − *φY * ≤ *C* · *nδ* · *dB (X , Y)*

with *X , Y* mono-filtrations and *δ* ∈ [0, 1); the proof of this state- ment is implicit in [[21](#_bookmark47), Thm 3]. With similar ideas, we show that the same result holds in the multi-parameter case.

**Theorem 4.** *If Ф is additive and there exists C >* 0 *and δ* ∈ [0, 1) *such that*

*δ*

**Proof.** Absolute boundedness ensures that the left-hand side is

well-defined by [Theorem 1](#_bookmark11). Now we use the definition of  · *L*2

 *ФX* − *ФY * *L*2

≤ *C* · *n* · *d*

*match*

*(X , Y)*

and the internal stability of *φ* to obtain

2

*for all bi-filtrations X and Y, then Ф is trivial.*

 *Ф* −*Ф*

*X*

 2 = ∫

¯*w(p, q)* · *φ*

*(λ , λ )*−*w(p, q)* · *φ*

*(λ , λ )*¯ *dμ*

**Proof.** Assume to the contrary that there exists a bi-filtration *X*

*Y*

*L*2

*X4*

*p*

*q*

*Y4*

*p*

*q*

such that  *ФX * *L*2 *>* 0. Then, writing *O* for the empty bi-filtration,

by additivity we get  *Ф*п*n X* − *ФO * *L*2 = *n * *ФX* − *ФO * *L*2 *>* 0. On

≤ ∫ *(w(p, q)* · *v*

2

· *n* · *d (X , Y )) dμ*

*n i*=1

*∆(*2*)*

*∆(*2*)*

3 *B 4 4*

the other hand, *dmatch(*п*i* 1 *X , O)* = *dmatch(X , O)*. Hence, with *C* and

*δ* as in the statement of the theorem,

=

= *(v*3 · *n)*2 ∫

*(w(p, q)* · *dB (X4, Y4 ))*2*dμ*

*Ф n X ФO L*2

*i*=1

п −

*i*=1

=  *n ФX* − *ФO L*2

*∆(*2*)*

= *n*1−*δ* − *L* →∞

*C* · *nδ* · *dmatch(*п*n X , O)*

*C* · *nδ* · *dmatch(X , O)*

Since *w(p, q)* is zero outside *R* × *R*, the integral does not change when restricted to *∆*(2) ∩ (*R* × *R*). Within this set, *w(p, q)* simpli-

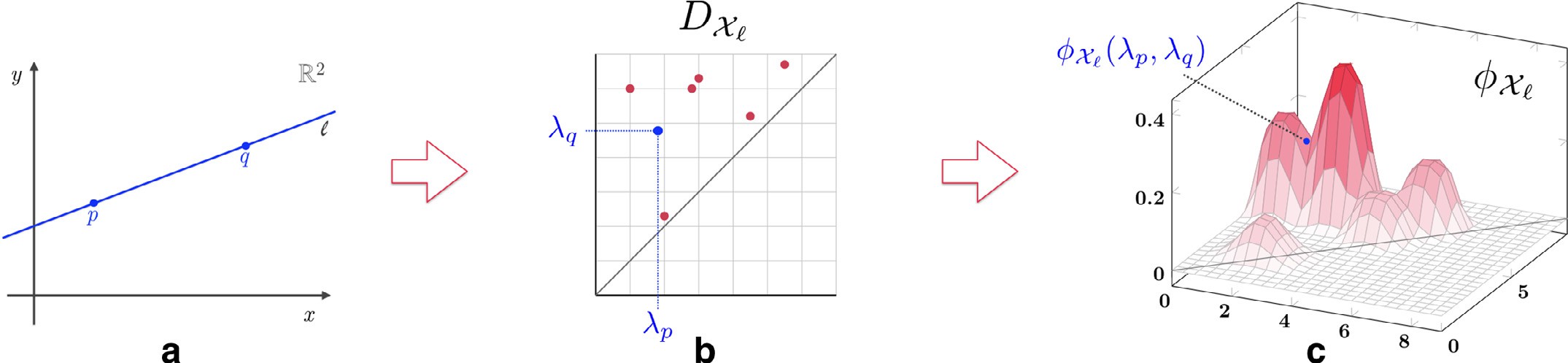
*ФX ФO* 2 *n*

*C* · *dmatch(X , O)* −→ ∞

*,*

fies to *4*ˆ*,* with *4* the line through *p* and *q*. Hence, we can further

a contradiction. ¤



**Fig. 4.** An illustration of the construction of a feature map for multi-parameter persistent homology. (a) Given a bi-filtration *X* and a point (*p*, *q*) ∈ *∆*(2), the line *4* passing through them is depicted and the parameter *λp* and *λq* computed. (b) The point (*λp* , *λq* ) is embedded in the persistence diagram of the mono-filtration *X4* obtained as the slice of *X* along *4*. (c) The point (*λp* , *λq* ) is assigned the value *фX4 (λp, λq )* via the feature map *ф*.

**a**



**b**

**Fig. 5.** (a) The two given slices realize the largest and smallest possible slope among all slices traversing the pink box pair. It can be easily seen that the difference of the unit vector of the center line to one of the unit vectors of these two lines realizes *A* for the given box pair. (b) Computing variations for the center slice and a traversing slice of a box pair.

##### Approximability

We provide an approximation algorithm to compute the kernel

pairs. That is,

#### *X , Y Ф*

( ⟩ = ∫

*∆(*2*)*

*ФX ФY dμ* =

*(B ,B )*

Σ

*∆(*2*)* ∩*(B*1 ×*B*2 *)*

*ФX ФY dμ.*

of two bi-filtrations *X* and *Y* up to any absolute error *‹ >* 0. Re- 1 2

call that our feature map *Ф* depends on the choice of a bounding box *R*. In this section, we assume *R* to be the unit square [0, 1] × [0, 1] for simplicity. We prove the following theorem that shows our kernel construction admits an eﬃcient approximation scheme that is polynomial in 1/*‹* and the size of the bi-filtrations.

∫

For each box pair, we compute an approximation interval for the integral, and sum them up using interval arithmetic to obtain *Js*.

We first give some (almost trivial) bounds for *X , Y Ф*. Let (*B*1, *B*2 ) be a box pair with centers located at *c*1 and *c*2 , respectively. By construction, vol*(B*1 × *B*2 *)* = 2−4*s*. By the absolute boundedness of *ф*, we have

( ⟩

µ

**Theorem 5.** *Assume ф is absolutely bounded, Lipschitz, internally*

*stable and efficiently computable. Given two bi-filtrations X and Y of size n and ‹ >* 0, *we can compute a number r such that r* ≤ (*X , Y*⟩*Ф* ≤

*∆* ∩*(B*1 ×*B*2 *)*

∫ *(*2*)*

*ФX ФY dμ* ≤

*(B*1 ×*B*2 *)*

∫

### 1 1

√2 1 √2

*v n* ·

*v*1*n*

¶*dμ* (7)

*r* + *‹ in polynomial time in n and* 1/*‹.*

*v*2 *n*2

1

*v*2 *n*2

1

The proof of [Theorem 5](#_bookmark18) will be illustrated in the following para-

= 2 vol*(B*1 × *B*2 *)* = 24*s*+1 *,* (8)

graphs, postponing most of the technical details to [Appendix A](#_bookmark24).

where 1*/*√2 is the maximal weight. Let *U* :

*v*2 *n*2

= 24*s*+1 . If *c*1 ≤ *c*2 , then

1

*Algorithm.* Given two bi-filtrations *X* and *Y* of size *n* and *‹ >* 0, our goal is to eﬃciently approximate *X , Y Ф* by some number *r*. On the highest level, we compute a sequence of approximation in- tervals (with decreasing lengths) *J*1*, J*2*, J*3*, . . . ,* each containing the desired kernel value *X , Y Ф*. The computation terminates as soon as we find some *Ji* of width at most *‹*, in which case we return the left endpoint as an approximation to *r*.

( ⟩

( ⟩

For *s* ∈ N (N being the set of natural numbers), we compute *Js*

we can choose [0, *U*] as approximation interval. Otherwise, if *c*1 /≤

*c*2 *,* then *∆(*2*)* ∩ *(B*1 × *B*2 *)* = ∅; we simply choose [0,0] as approxi- mation interval.

We can derive a second lower and upper bound for *X , Y Ф* as follows. We evaluate *ФX* and *ФY* at the pair of centers (*c*1 , *c*2), which is possible due to the eﬃciency hypothesis of *ф*. Let *vX* = *ФX (c*1 *, c*2 *)* and *vY* = *ФY (c*1 *, c*2 *)*. Then, we compute *varia- tionsδX , δY* ≥ 0 relative to the box pair, with the property that, for any pair (*p*, *q*) ∈ *B*1 × *B*2 , *ФX (p, q)* ∈ [*vX* − *δX , vX* + *δX* ]*,* and *ФY (p, q)* ∈ [*vY* − *δY , vY* + *δY* ]. In other words, variations describe

( ⟩

*X*

*Y*

1

as follows. We split *R* into 2*s* × 2*s* congruent squares (each of side

how far the value of *Ф* (or *Ф* ) deviates from its value at (*c* ,

length 2−*s*) which we refer to as *boxes*. See [Fig. 5](#_bookmark16)(a) for an example when *s* 3. We call a pair of such boxes a *box pair*. The integral from [(1)](#_bookmark5) can then be split into a sum of integrals over all 24*s* box

=

*c*2 ) within *B*1 × *B*2 . Combined with the derivations starting in [(7)](#_bookmark19), we have for any pair (*p*, *q*) ∈ *B*1 × *B*2 ,

max {0*, (vX* − *δX )(vY* − *δY )*} (9)

≤*ФX (p, q)ФY (p, q)* (10)

among all slices traversing the box pair. Using [Lemma 7](#_bookmark23), we see that

2*A* + *B*

#### *D ,*

≤

*M4*ˆ*c*

½ *n*

≤ min *, (vX* + *δX )(vY* + *δY )*¾*.* (11)

*v*2 2

1

2

and we set

*v*3*n(*2*A* + *B)*

By multiplying the bounds obtained in [(9)](#_bookmark20) by the volume of

*δ* := √2*M4*ˆ

*c*

+ *v*1*nW* + *v*2*nL.* (12)

*∆*(2) ∩ (*B*1 × *B*2 ), we get a lower and an upper bound for the integral of *ФX ФY* over a box pair (*B*1 , *B*2 ). By summing over all possible box pairs, the obtained lower and upper bounds are the endpoints of *Js*.

*Variations.* We are left with computing the variations relative to a box pair. For simplicity, we set *δ* := *δX* and explain the procedure only for *X* ; the treatment of *Y* is similar.

We say that a slice *4 traverses* (*B*1, *B*2) if it intersects both boxes in at least one point. One such slice is the *center slice 4c*, which is the slice through *c*1 and *c*2. See [Fig. 5](#_bookmark16)(b) for an illustration. We set *D* to be the maximal bottleneck distance of the center slice and every other slice traversing the box pair (to be more precise, of the persistence diagrams along the corresponding slices). We set *W* as the maximal difference between the weight of the center slice and any other slice traversing the box pair, where the weight *w* is de- fined as in [Section 3](#_bookmark12). Write *λc*1 for the parameter value of *c*1 along the center slice. For every slice *4* traversing the box pair and any point *p* ∈ *4* ∩ *B*1, we have a value *λp*, yielding the parameter value of *p* along *4*. We define *L*1 as the maximal difference of *λp* and *λc*1 among all choices of *p* and *4*. We define *L*2 in the same way for *B*2 and set *L* := max {*L*1 , *L*2 }. With these notations, we obtain [Lemma 6](#_bookmark22) below.

**Lemma 6.** *For all* (*p*, *q*) ∈ *B*1 × *B*2 ,

*v*3*n*

It follows from [Lemmas 6](#_bookmark22) and [7](#_bookmark23) that *δ* indeed satisfies the required

variation property.

We remark that *δ* might well be equal to ∞, if the box pair admits a traversing slice that is horizontal or vertical, in which case the lower and upper bounds derived from the variation are vacuous. While [(12)](#_bookmark21) looks complicated, the values *v*1*, v*2*, v*3 are constants coming from the considered feature map *ф*, and all the remaining values can be computed in constant time using elementary geometric properties of a box pair. We only explain the computation of *A* in [Fig. 5](#_bookmark16)(a) and skip the details of the other values.

*Analysis.* At this point, we have not made any claim that the algo- rithm is guaranteed to terminate. However, its correctness follows at once because *Js* indeed contains the desired kernel value. More- over, handling one box pair has a complexity that is polynomial in *n*, because the dominant step is to evaluate *ФX* at the center (*c*1, *c*2). Hence, if the algorithm terminates at iteration *s*0, its complex- ity is

*s*0

Σ ¡ ¢

*O* 24*s poly(n) .*

*s*=1

This is because in iteration *s*, 24*s* box pairs need to be consid- ered. Clearly, the geometric series above is dominated by the last iteration, so the complexity of the method is *O(*24*s*0 *poly(n))*. The last (and technically most challenging) step is to argue that *s*0 =

*‹*

|*ФX (p, q)* − *ФX (c*1*, c*2 *)*| ≤ √2 *D* + *v*1*nW* + *v*2*nL.*

*O(*log *n* + log 1 *),* which implies that the algorithm indeed termi- nates and its complexity is polynomial in *n* and 1/*‹*.

**Proof.** Plugging in [(6)](#_bookmark10) and using triangle inequality, we obtain

|*ФX (p, q)* − *ФX (c*1*, c*2 *)*|

¯

= ¯*4*ˆ*фX4 (λp, λq )* − *4*ˆ*c фX4c (λc*1 *, λc*2 *)*¯

≤ *4*ˆ¯*ф (λ , λ )* − *ф (λ , λ )*¯ + *ф (λ , λ )*¯*4*ˆ − *4*ˆ ¯

*X4*

*p*

*q*

*X4c*

*p*

*q*

*X4c*

*p*

*q*

*c*

To see that we can achieve any desired accuracy for the value of the kernel, i.e., that the interval width tends to 0, we observe that, if the two boxes *B*1, *B*2 are suﬃciently far away and the resolution *s* is suﬃciently large, the magnitudes *A*, *B*, *W*, and *L* in [(12)](#_bookmark21) are all small, because the parameterizations of two slices traversing the

box pair are similar (see [Lemmas 11](#_bookmark28)–[14](#_bookmark31) in [Appendix A](#_bookmark24)). Moreover,

+*4*ˆ*c фX4c (λp, λq )* − *фX4c (λc*1 *, λc*2 *)*

¯ ¯

and bound the three parts separately. The first summand is up- per bounded by *v*3 *nD* because of internal stability of the feature

√

2

if every slice traversing the box pair has a suﬃciently large weight (i.e., the slice is close to the diagonal), the value *M* in [(12)](#_bookmark21) is suﬃ- ciently large. These two properties combined imply that the varia- tion of such a box pair (which we refer to as the *good* type) tends

map *ф* and because *4*ˆ ≤ √1 for any slice *4*. The second sum-

2

to 0 as *s* goes to ∞. Hence, the bound based on the variation tends

mand is upper bounded by *v*1*nW* by the absolute boundedness of *ф*. The third summand is bounded by *v*2*nL,* because  *(λp, λq )* − *(λc*1 *, λc*2 *)* 2 ≤ *(λp, λq )* − *(λc*1 *, λc*2 *)* ∞ ≤ *L* and by *ф* being Lips-



2

√

√ 1

chitz, ¯*фX (λp, λq )* − *фX (λc , λc )*¯ ≤ 2*v*2*nL,* and *4*ˆ ≤ √ . The re-

*4c*

*4c*

1

2

2

to the correct value for good box pairs.

However, no matter how high the resolution, there are always *bad* box pairs for which either *B*1, *B*2 are close, or are far but close to horizontal and vertical, and hence yield a very large variation.

sult follows. ¤

Next, we bound *D* by simple geometric quantities. We use the following lemma, whose proof appeared in [[48]](#_bookmark57):

**Lemma 7.** [*[48]*](#_bookmark57) *Let 4 and 4*r *be two slices with parameterizations b* + *λa and b*r + *λa*r *, respectively. Then, the bottleneck distance of the two persistence diagrams along these slices is upper bounded by*



2 *a* − *a*r ∞ + *b* − *b*r ∞ *.*

*4*ˆ*4*ˆr

We define *A* as the maximal infinity distance of the directional vector of the center slice *4c* and any other slice *4* traversing the box pair. We define *B* as the maximal infinity distance of the base point of *4c* and any other *4*. Finally, we set *M* as the minimal weight

For each of these box pairs, the bounds derived from the variation

are vacuous, but we still have the trivial bounds [0, *U*] based on the absolute boundedness of *ф*. Moreover, the total volume of these bad box pairs goes to 0 when *s* goes to ∞ (see [Lemmas 9](#_bookmark25) and[10](#_bookmark29) in [Appendix A](#_bookmark24)). So, the contribution of these box pairs tends to 0. These two properties complete the proof of [Theorem 5](#_bookmark18).

A more careful investigation of our proof shows that the com- plexity of our algorithm is *O(n*80+*k (*1*/‹)*40 *),* where *k* is the eﬃ- ciency constant of the feature map ([Section 2](#_bookmark4)). We made little ef- fort to optimize the exponents in this bound.

##### Conclusions and future developments

We restate our main results for the case of a multi-filtration

*X* with *d* parameters: there is a feature map that associates to *X*

a real-valued function *ФX* whose domain is of dimension 2*d*, and introduces a kernel between a pair of multi-filtrations with a sta- ble distance function, where the stability bounds depend on the

*cne*1 *ue*2 ≤ *‹*

if and only if *s*

log *c*+*e*1 log *n*+log 1 . Hence, choosing

2

*‹*

≥ *e*

(2*d*-dimensional) volume of a chosen bounding box. The proofs of these generalized results carry over from the results of this paper. Moreover, assuming that *d* is a constant, we claim that the kernel can be approximated in polynomial time to any constant (with the

*s* :=

1

*e* = *O*

» log *c* + *e*1 log *n* + log ¼

*‹*

2

³log *n* + log 1

polynomial exponent depending on *d*). A proof of this statement requires to adapt the definitions and proofs of [Appendix A](#_bookmark24) to the higher-dimensional case; we omit details.

´

*‹*

Other generalizations include replacing filtrations of simplicial complexes with persistence modules (with a suitable finiteness condition), passing to sublevel sets of a larger class of (tame) func- tions and replacing the scale-space feature map with a more gen- eral family of single-parameter feature maps. All these generaliza- tions will be discussed in subsequent work.

The next step is an eﬃcient implementation of our kernel ap-

ensures that width*(Js*0 *)* ≤ *‹*. ¤

In the rest of this section, we will show that width*(Js )* =

*O(n*2 *u*0*.*1 *)*.

*Classifying box pairs.* For the analysis, we partition the box pairs considered by the algorithm into 4 disjoint classes. We call a box pair (*B*1, *B*2):

proximation algorithm. We have implemented a prototype in C++, realizing a more adaptive version of the described algorithm. We

* *null* if *c*1 *kc*2 ,
* *close* if *c*1 ≤ *c*2 such that  *c*1 − *c*2 2 *<*



√*u,*

√

have observed rather poor performance due to the sheer number of box pairs to be considered. Some improvements under consid-

* *non-diagonal* if *c*1 ≤ *c*2 such that *c*1 − *c*2 2 ≥

that traverses (*B*1 , *B*2 ) satisfies *4*ˆ *< u* 5 *,*

1

*u* and any line *4*

eration are to precompute all combinatorial persistence diagrams (cf. the barcode templates from [[44]](#_bookmark52)), to refine the search space adaptively using a quad-tree instead of doubling the resolution and to use techniques from numerical integration to handle real-world data sizes. We hope that an eﬃcient implementation of our kernel will validate the assumption that including more than a single pa- rameter will attach more information to the data set and improve the quality of machine learning algorithms using topological fea- tures.

##### Declaration of interests

The authors declare that they have no known competing finan- cial interests or personal relationships that could have appeared to influence the work reported in this paper.

##### Acknowledgments

This work was initiated at the Dagstuhl Seminar 17292 “Topol- ogy, Computation and Data Analysis”. We thank all members of the breakout session on multi-dimensional kernel for their valu- able suggestions. RC, UF, and MK acknowledge support by the Aus- trian Science Fund (FWF) grant P 29984-N35. CL is partially sup- ported by ARCES (University of Bologna), and FAR 2017 (University

* + *good* if it is of neither of the previous three types.

According to this notation, the integral from [(1)](#_bookmark5) can then be split as

(*X , Y*⟩*Ф* = (*X , Y*⟩*null* + (*X , Y*⟩*close* + (*X , Y*⟩*non*−*diag* + (*X , Y*⟩*good ,*

where, (*X , Y*⟩*null* is defined as *(B*1 *,B*2 *) null* .*∆(*2*)* ∩*(B*1 ×*B*2 *) ФX ФY dμ,* and analogously for the other ones. We let *Js*,*null* , *Js*,*close* , *Js,non*−*diag, Js*,*good* denote the four approximation intervals obtained from our

Σ

algorithm when summing up the contributions of the correspond- ing box pairs. Then clearly, *Js* is the sum of these four intervals. For simplicity, we will write *Jnull* instead of *Js*,*null* when *s* is fixed, and

likewise for the other three cases.

We observe first that the algorithm yields *Jnull* = [0*,* 0]*,* so null box pairs can simply be ignored. Box pairs that are either close or non-diagonal are referred to as *bad* box pairs in [Section 5](#_bookmark17). We proceed by showing that the width of *Jclose* , *Jnon*−*diag,* and *Jgood* are

all bounded by *O*(*n*2*u*0.1).

*Bad box pairs.* We start with bounding the width of *Jclose* . Let *B*close be the union of all close box pairs. Note that our algorithm assigns to each box pair (*B*1, *B*2) an interval that is a subset of [0,

of Modena and Reggio Emilia). BW is partially supported by NSF grant DBI-1661375, IIS-1513616 and NIH grant R01-1R01EB022876-

*U*]. Recall that *U*

*v*2 *n*2

= 24*s*+1 . *U* can be rewritten as

1

*v*2 *n*2

1 vol*(B*1 × *B*2 *),*

2

01. [Fig. 3](#_bookmark13) is generated by Lin Yan.

##### Appendix A. Details on the Proof of [Theorem 5](#_bookmark18)

*Overview.* Recall that our approximation algorithm produces an approximation interval *Js* for *s* ∈ N by splitting the unit square into

where vol(*B*1 × *B*2 ) is the 4-dimensional volume of the box pair (*B*1 ,

*B*2). It follows that

*v*2 *n*2

width*(Jclose )* ≤ 1 vol*(B*close *).* (A.1)

2

2*s* × 2*s* boxes. For notational convenience, we write *u* := 2−*s* for the side length of these boxes.

2

We would like to argue that the algorithm terminates after

**Lemma 9.** *For u* ≤ 1 *,* vol*(B*

close

*)* ≤ 4*πu.*

*O(*log *n* + log 1 *)* iterations, which means that after that many it- erations, an interval of width *‹* has been produced. The following [Lemma 8](#_bookmark27) gives an equivalent criterion in terms of *u* and *n*.

*‹*

**Proof.** Fixed a point *p* ∈ *R*, for each point *q* ∈ *R* such that *(p, q)* ∈ *B*close and *p < q*, there exists a unique close box pair (*B*1 , *B*2 ) that contains (*p*, *q*). By definition of close box pair, we have that:

**Lemma 8.** *Assume that there are constants e*1, *e*2 *>* 0, *such*

*that* width*(Js )* = *O(ne*1 *ue*2 *). Then,* width*(Js )* ≤ *‹ for some s*0 =  *p* − *q * 2 ≤  *p* − *c*1  2 + *c*1 − *c*2  2 + *c*2 − *q * 2 ≤

√*u* +

√2*u.*

## ¡ 1 ¢ 0

*O* log *n* + log *‹*

*.*

**Proof.** Assume that width*(Js )* ≤ *cne*1 *ue*2 for constants *c* and *s* suf- Moreover, for *u* ≤ 1 *,* √2*u* ≤ √*u,* and so  *p* − *q * 2 ≤ 2√*u*. Equiva-

2 √

ficiently large. Since *u* = 2−*s,* a simple calculation shows that lently, *q* belongs to the 2-ball *B(p,* 2 *u)* centered at *p* and of radius

2√*u*. Then,

based on the fact that

*4*ˆ ≥ *M*. The same bound holds for *δY* .

vol*(B*

close

*)* = ∫

1*dμ* ≤ ∫ ∫

√ 1*dμ*

Hence,

# µ µ ¶¶

close

∫

*p*

∈*R q*∈*B(p,*2 *u)*

width*(Jgood*

*)* = *O n*2

*A* + *B W L .*

*M*2

+ +

≤ *p*∈*R*

*B*

4*πudμ* = 4*πu.*

It remains to show that *A*+*B* + *W* + *L* = *O(u*0*.*1 *)*. Note that *M* ≥ *u* 1

*M*2 5

¤

because the box pair is assumed to be good. We will show in the

Consequently, combined with [(A.1)](#_bookmark26), we have

4*πv*2 *n*2

width*(Jclose )* ≤ 1 *u* = *O(n*2*u*0*.*1 *).*

2

Note that *u <* 1 and hence, *u* ≤ *u*0.1.

For the width of *Jnon*−*diag,* we use exactly the same reasoning, making use of the following [Lemma 10](#_bookmark29). Let *B*non-diag be the union of all non-diagonal box pairs.

next lemmas that *A*, *B*, *W*, and *L* are all in *O(*√*u),* proving that the

term is indeed in *O*(*u*0.1). This completes the proof of the complex- ity of the algorithm.

**Lemma 11.** *Let* (*B*1 , *B*2 ) *be a good box pair. Let a*, *a*r *be the unit direction vectors of two lines that pass through the box pair. Then,*

 *a* − *a*r  ∞ ≤ 2√*u. In particular, A* = *O(*√*u).*

**Proof.** Since (*B*1, *B*2) is a good box pair, the largest value for

5

**Lemma 10.** *For u* ≤ 2− 2 *,* vol*(B*

non-diag

*)* ≤ √

1

2*u* 5 *.*

 *a* − *a*r  ∞ is achieved when *4* and *4*r correspond to the lines pass- ing through the box pair(*B*1, *B*2) with minimum and maximum

**Proof.** Fixed a point *p* ∈ *R*, for each point *q* ∈ *R* such that *(p, q)* ∈

*B*non-diag and *p < q*, there exists a unique non-diagonal box pair (*B*1 ,

*B*2) that contains (*p*, *q*). We have that *q* lies in:

slope, respectively. By denoting as *c*1 = *(c*1*,x, c*1*,y ), c*2 = *(c*2*,x, c*2*,y )* the centers of *B*1, *B*2, we define *4* to be the line passing through the points *c*1 + *(*− *u , u ), c*2 + *( u ,* − *u )*. Similarly, let us call *4*r the

* line passing through the points *c*1 + *( u ,* − *u ), c*2 + *(*− *u , u )*. So, the

2

2

2

2

Triangle *T*1 (*p*) of vertices *p* = *(p*1 *, p*2 *),* (1, *p*2 ), and *(*1*, p*2 + *(*1 −

2 2 2 2

*p*1 *) a*2 *),* if the line *4* of maximum slope passing through *B*1 × *B*2

*a*1

is such that *4*ˆ = *a*2 where *a* = *(a*1 *, a*2 *)* is the (positive) unit di-

unit direction vector *a* of *4* can be expressed as

*(c*2 + *( u ,* − *u ))* − *(c*1 + *(*− *u , u ))*

2

2

2

*a* = ¨ 2 2

2 2 ¨ *.*

rection vector of *4*;

* Triangle *T* (*p*) of vertices *p* = *(p , p ),* (*p* , 1), and *(p*

+ *(*1 −

2

2

¨*(c*2 + *( u ,* − *u ))* − *(c*1 + *(*− *u , u ))*¨

2

1

2

1

1

*p*2 *) a*1 *,* 1*),* if the line *4* of minimum slope passing through

*a*

2

*B*1 × *B*2 is such that *4*ˆ = *a*1 where *a* = *(a*1 *, a*2 *)* is the (positive)

Similarly, the unit direction vector *a*r of *4*r is described by

r *(c*2 + *(*− *u , u ))* − *(c*1 + *( u ,* − *u ))*

unit direction vector of *4*.

*a* = ¨ 2 2

2 2 ¨ *.*

¨*(c*2 + *(*− *u , u ))* − *(c*1 + *( u ,* − *u ))*¨

2

2

2

2

2

Let us bound the area of the two triangles. Since the calcula-

tions are analogous, let us focus on *T*1(*p*). By definition, the basis

Then, by denoting as ( · , · ⟩ the scalar product,

of *T*1 (*p*) is smaller than 1 while its height is bounded by *a*2 . The

 *a* − *a*r  2 ≤ *a* − *a*r  2 = *a * 2 + *a*r  2 − 2(*a, a*r⟩ = 2*(*1 − (*a, a*r⟩*)*

maximum value for the height of *T* (*p*) is achieved for *a*

*a*1

= 1 µ



*c*2 −*c*1 + *(u,* −*u)*

*c*2 −*c*1 + *(u,* −*u)* 2



*c*2 −*c*1 + *(*−*u, u)*

*c*2 −*c*1 + *(*−*u, u)* 2

∞

2

2

2

# ¿ À¶

1

by exploiting the identity *a*2 + *a*2 = 1*,* we have

2 *u* 5 . So,

= 2 1− *,*

1 2

³ *a*2 ´2 *u* 5

2

µ   *c* − *c*

— 2*u * ¶

*a*1 =

2 *.*

1 − *u* 5

5 2

2

2 1 2 1 2 *.*

*c*2 − *c*1 + *(u,* −*u) * 2 *c*2 − *c*1 + *(*−*u, u)* 2



2 2

—

=

Under the conditions *u* ≤ 2− 2 and 1 *u*− 5 ≥ 1 we have

By an elementary calculation, one can prove that

*a*2 √

≤

#### *a*



 1

2*u* 5 *.*

 *c*2 − *c*1

+ *(u,* −*u) * 2  *c*2

— *c*1

+ *(*−*u, u) * 2

1

√2 1

√2 1

= ,4*u*2 ¡*u*2 + 2*(c*2*,x* − *c*1*,x )(c*2*,y* − *c*1*,y )*¢

+ *c*2 − *c*1  *.*

4

Therefore, area*(T*1 *(p))* ≤

nally,

2 *u* 5 *.* Similarly, area*(T*2 *(p))* ≤

2 *u* 5 *.* Fi-

2

Then,

vol*(B*non-diag *)* = ∫

≤ ∫

∫

*B*non-diag

∫

1*dμ*

1*dμ*

 *a* − *a*r  2

⎛

∞

### ≤ 2⎝1 −

⎞

 ⎠

*p R q*∈*T*1 *(p)*∪*T*2 *(p)*

∈

√

1

√

1

≤ *p*∈*R*

2*u* 5 *dμ* ≤

2*u* 5 *.*

¤ = 2



2

2

,

4*u u* + 2*(c* − *c )(c* − *c )* + *c* − *c*

2 2

¡

*c*2 − *c*1 2 − 2*u*

2*,x* 1*,x* 2*,y* 1*,y*

¢

2 1

4

2

⎛

⎝1 +

⎞

 ⎠*.*

*Good box pairs.* For good box pairs, we use the fact that the

variation of a box pair yields a subinterval of [*(vX* − *δX )(vY* − *δY )*vol*(B*1 × *B*2 *), (vX* + *δx )(vY* + *δY )*vol*(B*1 × *B*2 *)*] as an approxima- tion, so the width is bounded by 2*(vX δY* + *vY δX )*vol*(B*1 × *B*2 *)*. Let

Since (*B*1,

2

⎛

4¡*u*2 + 2*(c*

2 − *c*1  2 ≥

*u*. So,

⎞

*)*¢ + 1

*B*good be the union of all good box pairs. Since the volumes of all



,

4*u u* + 2*(c* − *c )(c* − *c )* + *c*

2 2

¡

2*u* − *c* − *c*

2

2 1

2

2

2*,x* 1*,x* 2*,y* 1*,y*

¢

2

— *c*

1

4

2

*B* ) is a good box pair,  *c*

√

∞

*a* − *a*r  2

≤ 2⎝1 + ,

2*u* − 1 ⎠*.*

— *c*

— *c*

2*,x*

1*,x*

*)(c*

2*,y*

1*,y*

absolute boundedness, *vX* and *vY* are in *O*(*n*), and recall that by definition, Since

good box pairs sum up to at most one, that is, vol*(B*good *)* ≤ 1*,* it follows that the width of *Jgood* is bounded by 2*(vX δY* + *vY δX )*. By

∞

,4¡*u*2 + 2*(c*2*,x* − *c*1*,x )(c*2*,y* − *c*1*,y )*¢

+ 1 ≥ 1*,* we have that

*δ* = *v*3 *n(*2*A* + *B)* + *v nW* + *v nL* = *O*µ*n*µ *A* + *B* + *W* + *L*¶¶

*X*

√2*M4*ˆ*c*

1

2

*M*2

 *a* − *a*r  2 ≤ 2*(*1 + 2*u* − 1*)* = 4*u.*

Therefore,

 *a* − *a*r  ∞ ≤ 2√*u.*

Finally,

¨*b* − *b*r¨∞

¤

≤ 4 *u* = 4√*u.*

¤

√*u*

**Lemma 12.** *Let* (*B*1 , *B*2 ) *be a good box pair. Let 4* = *aλ* + *b, 4*r =

*a*r *λ* + *b*r *be two lines that pass through the box pair such that a*, *a*r *are*

*unit direction vectors and b*, *b*r *are the intersection points with the di- agonal of the second and the fourth quadrant. Then b* − *b*r ∞ ≤ 4√*u. In particular, B* = *O(*√*u).*

**Proof.** Since (*B*1, *B*2) is a good box pair, the largest value for  *b* −

*b*r ∞ is achieved when *4* and *4*r correspond to the lines passing



**Lemma 13.** *Let* (*B*1 , *B*2 ) *be a good box pair. Let 4*ˆ*, 4*ˆr *be the weights of two lines 4 and 4*r *that pass through the box pair. Then 4*ˆ *4*ˆ 4√*u. In particular, W* = *O(*√*u).*

**Proof.** If *4*ˆ = *a*1 and *4*ˆr = *a*r1 *,* then, by applying [Lemma 11](#_bookmark28),

| − r| ≤

|*4*ˆ − *4*ˆr | = |*a* − *a*r | ≤ ¨*a* − *a*r¨ ≤ 2√*u.*

1

1

∞

through the box pair(*B*1, *B*2) with minimum and maximum slope,

On the other hand, if *4*ˆ = *a* and *4*ˆr = *a*r *,* then there exists a line *4*rr

respectively. By denoting the centers of *B*1 and *B*2 by *c*1 and *c*2 , we 1 √ √

2

2

define *4* to be the line passing through the points *c* + *(*− *, ),*

*u u*

1

2

2

2

passing through the box pair (*B*1 , *B*2 ) such that *a*rr = *(* 2 *,*

2 *)*. By

*c*2 + *( u ,* − *u )*. Similarly, let us call *4*r the line passing through the

2

2

applying twice [Lemma 11](#_bookmark28),

points *c*1 + *( u ,* − *u ), c*2 + *(*− *u , u )*. So, *4* can be expressed as √ √

1 1

*c*2 + *( u ,* − *u )* − *c*1 − *(*− *u , u )*

2

2

2 2 2 2

³ ´ |*4*ˆ − *4*ˆr | = |*a* − *a*r | ≤ |*a* − 2 | + | 2 − *a*r |

*(x, y)* = ¨ 2 2 2 2 ¨ *t* + *c*1 + − *, ,*

*u*

*u*

2

2

rr rr r ¨

rr¨ ¨ rr r¨

¨*c*2 + *( u ,* − *u )* − *c*1 − *(*− *u , u )*¨ 2 2

2

2

2

2

2

√

where *t* is a parameter running on R. By intersecting *4* with the

+

= |*a*1 − *a*1 | + |*a*2 − *a*2 | ≤ *a* − *a* ∞ *a* − *a* ∞

≤ 4

*u.*

line *y* = −*x,* we get:

*u*

The cases *4*ˆ = *a*2 *, 4*ˆr = *a*r and *4*ˆ = *a*2 *, 4*ˆr = *a*r

can be treated analo-

2 1

*u*

*c* + − *c* + *u* ¤

2*,x* 2 1*,x* 2

gously to the previous ones.

¨*c* + *( ,* − *)* − *c*

*u u*

*( , )*¨ 2

2

2

2

*u u t* + *c*1*,x* −

− −

1

2

2

2

**Lemma 14.** *Let* (*p*, *q*), (*p*r, *q*r) *be two points in a good box pair*

−*c*2*,y* + + *c*1*,y* + *t* − *c* − *u ,*

*u u*

=

¨

*c*2 + *( u ,* − *u )* − *c*1 − *(*− *u , u )*

2

2 2

¨

1*,y*

2

2

2

2

2

(*B*1 , *B*2 ) *and let 4*, *4*r *be the lines passing through p*, *q and p*r, *q*r,

*respectively. In accordance with the usual parametrization, we have*

√

√

√

√

*that* |*λp* − *λp*r | ≤√ 2*u* + 4

*u and* |*λq* − *λq*r | ≤

2*u* + 4

*u. As a con-*

which can be written as

*c*1*,x* + *c*1*,y* − *c*2*,x* − *c*2*,y*

*sequence, L* = *O( u).*

*c*1*,x* + *c*1*,y* =

¨*c*

2 +

2 *,* − 2

—

1 −

— 2 *,* 2

*( u u ) c*

*t,*

*( u u )*¨

2

**Proof.** Thanks to the definition of *λp*, the triangular inequality and

[Lemma 12](#_bookmark30), we have that:

letting us deduce that

*λp* =  *p* − *b * 2 ≤ ¨*p* − *p*r¨2 + ¨*p*r − *b*r¨2 + ¨*b*r − *b*¨2

*(c*1*,x* + *c*1*,y )*¨*c*2 + *( u ,* − *u )* − *c*1 − *(*− *u , u )*¨

*t* = 2 2 2 2 2 *.*

≤ √2*u* + *λp*r + 4√*u.*

*c*1*,x* + *c*1*,y* − *c*2*,x* − *c*2*,y*

So, we have that *λ* − *λ*

≤ √2*u* + 4√*u,* and, similarly, *λ*

— *λ* ≤

So, by replacing *t* in the equation of *4* we retrieve *b*:

*c*2 + *( u ,* − *u )* − *c*1 − *(*− *u , u )*

¨*c*2 + *( u ,* − *u )* − *c*1 − *(*− *u , u )*¨

√2*u* + 4

√*u.* Then,

√

*p p*r

√

*p*r *p*

*b* = ¨ 2 2 2 2 ¨

Analogously, it can be proven that

2

2

2

2

2

|*λp* − *λp*r | ≤

2*u* + 4 *u.*

*(c*1*,x* + *c*1*,y )*¨*c*2 + *( u ,* − *u )* − *c*1 − *(*− *u , u )*¨

2 2 2 2 2

+ ³− *u , u* ´

|*λq* − *λq*r | ≤

√2*u* + 4√*u.*

*c*1*,x* + *c*1*,y* − *c*2*,x* − *c*2*,y*

### 1 2 2

+ *c*

*(u,* −*u)(c*1*,x* + *c*1*,y )*  *(c*2 − *c*1 *)(c*1*,x* + *c*1*,y )* ¤

= *c*1*,x* + *c*1*,y* − *c*2*,x* − *c*2*,y* + *c*1*,x* + *c*1*,y* − *c*2*,x* − *c*2*,y*

##### References

+*c* + ³− *u , u* ´*.*

### 1 2 2

Similarly,

*b*r *(*−*u, u)(c*1*,x* + *c*1*,y )*  *(c*2 − *c*1 *)(c*1*,x* + *c*1*,y )*

= +

*c*1*,x* + *c*1*,y* − *c*2*,x* − *c*2*,y c*1*,x* + *c*1*,y* − *c*2*,x* − *c*2*,y*

+*c* + ³ *,* − ´*.*

*u u*

### 1 2 2

So,

1. [Carstens CJ, Horadam KJ. Persistent homology of collaboration networks. Math Probl Eng 2013;2013:7.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0001)
2. [Chan JM, Carlsson G, Rabadan R. Topology of viral evolution. Proc Natl Acad Sci USA 2013;110(46):18566–71.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0002)
3. [Giusti C, Pastalkova E, Curto C, Itskov V. Clique topology reveals intrin- sic geometric structure in neural correlations. Proc Natl Acad Sci USA 2015;112(44):13455–60.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0003)
4. [Guo W, Banerjee AG. Toward automated prediction of manufacturing produc- tivity based on feature selection using topological data analysis. In: Proceed- ings of the IEEE international symposium on assembly and manufacturing; 2016. p. 31–6.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0004)

¨ ¨ ¨³

*c* + *c* ´ ¨

1. [Hiraoka Y, Nakamura T, Hirata A, Escolar EG, Matsue K, Nishiura Y. Hierarchical](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0005)

¨*b* − *b*r¨

∞

¨

= ¨ 2

1*,x*

1*,y*

— 1 *(u,* −*u)*¨

[structures of amorphous solids characterized by persistent homology. Proc Natl](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0005)

[Acad Sci USA 2016;113(26):7035–40.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0005)

[6] [Li L, Cheng WY, Glicksberg BS, Gottesman O, Tamler R, Chen R, et al. Identi-](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0006)

= ¯ *c*1*,x* + *c*1*,y* + *c*2*,x* + *c*2*,y* ¯ 

*c*1*,x* + *c*1*,y* − *c*2*,x* − *c*2*,y*

¨∞

*c*

+ *c*

— *c*

4*r*

— *c*

¯

∞

2*,x*

2*,y*

1*,x*

1*,y*

*(u,* −*u) *

[fication of type 2 diabetes subgroups through topological analysis of patient](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0006)

[similarity. Sci Transl Med 2015;7(311):311ra174.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0006)

[7] [Wang Y, Ombao H, Chung MK. Topological data analysis of single-trial elec- troencephalographic signals. Ann Appl Stat 2017;12(3):1506–34.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0007)

≤ |*c*

2*,x*

+ *c*2*,y*

* *c*1*,x*

*u.*

* *c*1*,y*|

1. [Yoo J, Kim EY, Ahn YM, Ye JC. Topological persistence vineyard for dynamic functional brain connectivity during resting and gaming stages. J Neurosci Methods 2016;267(15):1–13.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0008)

Since (*B*1 , *B*2 ) is a good box pair,

1. [Nicolau M, Levine AJ, Carlsson G. Topology based data analysis identifies a sub-](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0009)

*c*2*,x*

+ *c*2*,y*

— *c*1*,x*

— *c*1*,y*

= *c*2

— *c*1  1

≥ *c*2

— *c*1  2

≥ √*u.*

[group of breast cancers with a unique mutational profile and excellent sur- vival. Proc Natl Acad Sci USA 2011;108(17):7265–70.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0009)

1. [Turner K, Mukherjee S, Boyer DM. Persistent homology transform for modeling shapes and surfaces. Inf Inference J IMA 2014;3(4):310–44.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0010)
2. [Li C, Ovsjanikov M, Chazal F. Persistence-based structural recognition. In: Pro- ceedings of the ieee conference on computer vision and pattern recognition; 2014. p. 2003–10.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0011)
3. [Biasotti S, Falcidieno B, Spagnuolo M. Extended Reeb graphs for surface under- standing and description. In: Proceedings of the ninth international conference on discrete geometry for computer imagery; 2000. p. 185–97.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0012)
4. [Skraba P, Ovsjanikov M, Chazal F, Guibas L. Persistence-based segmentation of deformable shapes. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition – Workshop on non-rigid shape analysis and deformable image alignment; 2010. p. 45–52.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0013)
5. [Munkres JR. Elements of algebraic topology. Redwood City, CA, USA: Addis- on-Wesley; 1984.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0014)
6. [Edelsbrunner H, Letscher D, Zomorodian AJ. Topological persistence and sim- plification. Discr Comput Geom 2002;28:511–33.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0015)
7. [Zomorodian A, Carlsson G. Computing persistent homology. Discr Comput Geom 2005;33(2):249–74.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0016)
8. [Edelsbrunner H, Harer J. Persistent homology – a survey. Contemp Math 2008;453:257–82.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0017)
9. [Edelsbrunner H, Harer J. Computational topology: An introduction. Providence, RI, USA: American Mathematical Society; 2010.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0018)
10. [Edelsbrunner H, Morozov D. Persistent homology: Theory and practice. In: Pro- ceedings of the European congress of mathematics. European Mathematical Society Publishing House; 2012. p. 31–50.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0019)
11. [Cohen-Steiner D, Edelsbrunner H, Harer J. Stability of persistence diagrams. Discr Comput Geom 2007;37(1):103–20.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0020)
12. [Reininghaus J, Huber S, Bauer U, Kwitt R. A stable multi-scale kernel for topo- logical machine learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 4741–8.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0021)
13. [Bubenik P. Statistical topological data analysis using persistence landscapes. J Mach Learn Res 2015;16(1):77–102.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0022)
14. [Kwitt R, Huber S, Niethammer M, Lin W, Bauer U. Statistical topological data analysis – a kernel perspective. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, editors. Proceedings of the advances in neural infor- mation processing systems 28. Curran Associates, Inc.; 2015. p. 3070–8.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0023)
15. [Kusano G, Fukumizu K, Hiraoka Y. Persistence weighted Gaussian kernel for topological data analysis 2016;48:2004–13.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0024)
16. [Hofer C, Kwitt R, Niethammer M, Uhl A. Deep learning with topological signa- tures. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, editors. Proceedings of the advances in neural information processing systems](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0025)

[30. Curran Associates, Inc.; 2017. p. 1634–44.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0025)

1. [Carlsson G, Zomorodian A. The theory of multidimensional persistence. Discr Comput Geom 2009;42(1):71–93.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0026)
2. [Carlsson G, Mémoli F. Multiparameter hierarchical clustering methods. In: Lo- carek-Junge H, Weihs C, editors. Classification as a tool for research. Springer Berlin Heidelberg; 2010. p. 63–70.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0027)
3. [Chazal F, Cohen-Steiner D, Guibas LJ, Mémoli F, Oudot SY. Gromov-Haus- dorff stable signatures for shapes using persistence. In: Proceedings of the Eurographics symposium on geometry processing, 28; 2009. p. 1393–1403.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0028)
4. [Singh G, Mémoli F, Carlsson G. Topological methods for the analysis of high dimensional data sets and 3D object recognition. In: Proceedings of the Euro- graphics symposium on point-based graphics; 2007. p. 91–100.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0029)
5. [Sun J, Ovsjanikov M, Guibas L. A concise and provably informative multi-scale signature based on heat diffusion. Comput Graph Forum 2009;28(5):1383–92.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0030)
6. [Edelsbrunner H, Harer J. Jacobi sets of multiple Morse functions. In: Cucker F, DeVore R, Olver P, Süli E, editors. Proceedings of the foundations of computa- tional mathematics. Cambridge University Press; 2002. p. 37–57.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0031) [Minneapolis](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0031)
7. [Edelsbrunner H, Harer J, Patel AK. Reeb spaces of piecewise linear mappings. In: Proceedings of the twenty-forth annual symposium on computational ge- ometry; 2008. p. 242–50.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0032)
8. [Carriére M, Cuturi M, Oudot S. Sliced Wasserstein kernel for persistence dia- grams. In: Proceedings of the thirty-forth international conference on machine learning, 70; 2017. p. 664–73.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0033)
9. [Adams H, Chepushtanova S, Emerson T, Hanson E, Kirby M, Motta F, et al. Per- sistence images: a stable vector representation of persistent homology. J Mach Learn Res 2017;18(1):218–52.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0034)
10. [Adcock A, Rubin D, Carlsson G. Classification of hepatic lesions using the matching metric. Comput Vis Image Understand 2014;121:36–42.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0035)
11. [Di Fabio B, Ferri M. Comparing persistence diagrams through complex vectors. In: M V, P E, editors. Proceedings of the international conference on image analysis and processing. Springer, Cham; 2015. p. 294–305.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0036)
12. [Adcock A, Carlsson E, Carlsson G. The ring of algebraic functions on persistence bar codes. Homol Homot Appl 2016;18:381–402.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0037)
13. [Kališnik S. Tropical coordinates on the space of persistence barcodes. Found Comput Math 2018;19(1):1–29.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0038)
14. [Lesnick M. The theory of the interleaving distance on multidimensional persis- tence modules. Found Comput Math 2015;15(3):613–50.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0039)
15. Bjerkevik HB, Botnan MB, Kerber M. Computing the interleaving distance is NP-hard. arXiv:1811.09165; 2018.
16. [Dey TK, Xin C. Computing bottleneck distance for 2-d interval decomposable modules. In: Proceedings of the thirty-forth international symposium on com- putational geometry; 2018. p. 32:1–32:15.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0040)
17. [Biasotti S, Cerri A, Frosini P, Giorgi D. A new algorithm for computing the 2-dimensional matching distance between size functions. Pattern Recogn Lett 2011;32(14):1735–46.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0041)
18. [Kerber M, Lesnick M, Oudot S. Exact computation of the matching distance on 2-parameter persistence modules. In: Proceedings of the thirty-fifth interna- tional symposium on computational geometry; 2019. p. 46:1–46:15.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0042)
19. Lesnick M, Wright M. Interactive visualization of 2-D persistence modules. arXiv:1512.00180; 2015.
20. [Oudot S. Persistence theory: From quiver representation to data analysis. Mathematical surveys and monographs, 209. American Mathematical Society; 2015.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0043)
21. [Cohen-Steiner D, Edelsbrunner H, Harer J. Stability of persistence diagrams. Discr Comput Geom 2007;37(1):103–20.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0044)
22. [Biasotti S, Cerri A, Frosini P, Giorgi D, Landi C. Multidimensional size functions for shape comparison. J Math Imaging Vis 2008;32(2):161.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0045)
23. [Landi C. The rank invariant stability via interleavings. In: Chambers EW, Fasy BT, Ziegelmeier L, editors. Research in computational topology, 13. Springer; 2018. p. 1–10.](http://refhub.elsevier.com/S2590-1486(19)30005-6/sbref0046)