Journal of Machine Learning Research 18 (2017) 1-35 Submitted 7/16; Published 2/17

Persistence Images: A Stable Vector Representation of Persistent Homology

Henry Adams ADAMS@MATH.COLOSTATE.EDU
Tegan Emerson EMERSON@MATH.COLOSTATE.EDU
Michael Kirby KIRBY@MATH.COLOSTATE.EDU
Rachel Neville NEVILLE@MATH.COLOSTATE.EDU
Chris Peterson PETERSON@MATH.COLOSTATE.EDU
Patrick Shipman SHIPMAN@MATH.COLOSTATE.EDU
Department of Mathematics
Colorado State University
1874 Campus Delivery
Fort Collins, CO 80523-1874

Sofya Chepushtanova SOFYA.CHEPUSHTANOVA@WILKES.EDU
Department of Mathematics and Computer Science
Wilkes University
84 West South Street
Wilkes-Barre, PA 18766, USA

Eric Hanson ERIC.HANSON@TCU.EDU
Department of Mathematics
Texas Christian University
Box 298900
Fort Worth, TX 76129

Francis Motta MOTTA@MATH.DUKE.EDU
Department of Mathematics
Duke University
Durham, NC 27708, USA

Lori Ziegelmeier LZIEGEL1@MACALESTER.EDU
Department of Mathematics, Statistics, and Computer Science
Macalester College
1600 Grand Avenue
Saint Paul, MN 55105, USA

Editor: Michael Mahoney

Abstract

Many data sets can be viewed as a noisy sampling of an underlying space, and tools from topological data analysis can characterize this structure for the purpose of knowledge discovery. One such tool is persistent homology, which provides a multiscale description of the homological features within a data set. A useful representation of this homological information is a *persistence diagram* (PD). Efforts have been made to map PDs into spaces with additional structure valuable to machine learning tasks. We convert a PD to a finite-dimensional vector representation which we call a *persistence image* (PI), and prove the stability of this transformation with respect to small perturbations in the inputs. The

discriminatory power of PIs is compared against existing methods, showing significant performance gains. We explore the use of PIs with vector-based machine learning tools, such as linear sparse support vector machines, which identify features containing discriminating topological information. Finally, high accuracy inference of parameter values from the dynamic output of a discrete dynamical system (the *linked twist map*) and a partial differential equation (the *anisotropic Kuramoto-Sivashinsky equation*) provide a novel application of the discriminatory power of PIs.

Keywords: topological data analysis, persistent homology, persistence images, machine learning, dynamical systems

1. Introduction

In recent years, the field of topology has grown to include a large set of computational tools (Edelsbrunner and Harer, 2010). One of the fundamental tools is persistent homology, which tracks how topological features appear and disappear in a nested sequence of topological spaces (Edelsbrunner and Harer, 2008; Zomorodian and Carlsson, 2005). This multiscale information can be represented as a *persistence diagram* (PD), a collection of points in the plane where each point (x, y) corresponds to a topological feature that appears at scale x and disappears at scale y. We say the feature has a persistence value of $y - x$. This compact summary of topological characteristics by finite multi-sets of points in the plane is responsible, in part, for the surge of interest in applying persistent homology to the analysis of complex, often high-dimensional data. Computational topology has been successfully applied to a broad range of data-driven disciplines (Perea and Harer, 2013; Dabaghian et al., 2012; Chung et al., 2009; Heath et al.; Singh et al., 2008; Topaz et al., 2015; Pearson et al., 2015).

Concurrent with this revolution in computational topology, a growing general interest in data analysis has driven advances in data mining, pattern recognition, and machine learning (ML). Since the space of PDs can be equipped with a metric structure (*bottleneck* or *Wasserstein* (Mileyko et al., 2011; Turner et al., 2014)), and since these metrics reveal the stability of PDs under small perturbations of the data they summarize (Cohen-Steiner et al., 2007, 2010; Chazal et al., 2014), it is possible to perform a variety of ML techniques using PDs as a statistic for clustering data sets. However, many other useful ML tools and techniques (e.g., support vector machines (SVM), decision tree classification, neural networks, feature selection, and dimension reduction methods) require more than a metric structure. In addition, the cost of computing the bottleneck or Wasserstein distance grows quickly as the number of off-diagonal points in the diagrams increases (Di Fabio and Ferri, 2015). To resolve these issues, considerable effort (which we review in §2) has been made to map PDs into spaces which are suitable for other ML tools (Bubenik, 2015; Reininghaus et al., 2015; Rouse et al., 2015; Adcock et al., 2016; Donatini et al., 1998; Ferri et al., 1997; Chung et al., 2009; Pachauri et al., 2011; Bendich et al., 2016; Chen et al., 2015; Carrière et al., 2015; Di Fabio and Ferri, 2015). With the benefits and drawbacks of these approaches in mind, we pose the following question:

Problem Statement: How can we represent a persistence diagram so that

(i) the output of the representation is a vector in $\mathbb{R}^n$,

(ii) the representation is stable with respect to input noise,

(iii) the representation is efficient to compute,

(iv) the representation maintains an interpretable connection to the original PD, and

(v) the representation allows one to adjust the relative importance of points in different regions of the PD?

The main contribution of this paper is to study a finite-dimensional-vector representation of a PD called a *persistence image* (PI). We first map a persistence diagram B to an integrable function $\rho_B \colon \mathbb{R}^2 \to \mathbb{R}$ called a *persistence surface*. The surface ρ_B is defined as a weighted sum of Gaussian functions,[1] one centered at each point in the PD. The idea of persistence surfaces has appeared even prior to the development of persistent homology, in Donatini et al. (1998) and Ferri et al. (1997). Taking a discretization of a subdomain of ρ_B defines a grid. A persistence image, i.e., a matrix of pixel values, can be created by computing the integral of ρ_B on each grid box. This PI is a "vectorization" of the PD, and provides a solution to the problem statement above.

Criterion (i) is the primary motivation for developing PIs. A large suite of ML techniques and statistical tools (means and variances) already exist to work with data in $\mathbb{R}^n$. Additionally, such a representation allows for the use of various distance metrics (p-norms and angle based metrics) and other measures of (dis)similarity. The remaining criteria of the problem statement (ii-v) further ensure the usefulness of this representation.

The desired flexibility of (v) is accomplished by allowing one to build a PI as a weighted sum of Gaussians, where the weightings may be chosen from a broad class of weighting functions.[2] For example, a typical interpretation is that points in a PD of high persistence are more important than points of low persistence (which may correspond to noise). One may therefore build a PI as a weighted sum of Gaussians where the weighting function is non-decreasing with respect to the persistence value of each PD point. However, there are situations in which one may prefer different measures of importance. Indeed, Bendich et al. (2016) find that, in their regression task of identifying a human brain's age from its arterial geometry, the points of medium persistence (not high persistence) best distinguish the data. In such a setting, one may choose a weighting function with largest values for the points of medium persistence. In addition, the Homology Inference Theorem (Cohen-Steiner et al., 2007) states that when given a sufficiently dense finite sample from a space X, the points in the PD with sufficiently small birth times (and sufficiently high persistence) recover the homology groups of the space; hence one may choose a weighting function that emphasizes points near the death-axis and away from the diagonal, as indicated in the leftmost yellow rectangle of Figure 2.4 in Bendich (2009). A potential disadvantage of the flexibility in (v) is that it requires a choice; however, prior knowledge of one's particular problem may inform that choice. Moreoever, our examples illustrate the effectiveness of a standard choice of weighting function that is non-decreasing with the persistence value.

The remainder of this article is organized as follows. Related work connecting topological data analysis and ML is reviewed in §2, and §3 gives a brief introduction to persistent homology, PDs from point cloud data, PDs from functions, and the bottleneck and Wasserstein metrics. PIs are defined in §4 and their stability with respect to the 1-Wasserstein distance

1. In general, ρ_B can be a weighted sum of probability density functions
2. Weighting functions are restricted only to the extent necessary for our stability results in §5.

between PDs is proved in §5. Lastly, §6 contains examples of ML techniques applied to PIs generated from samples of common topological spaces, an applied dynamical system modeling turbulent mixing, and a partial differential equation describing pattern formation in extended systems driven far from equilibrium. Our code for producing PIs is publicly available at `https://github.com/CSU-TDA/PersistenceImages`.

2. Related Work

The space of PDs can be equipped with the bottleneck or Wasserstein metric (defined in §3), and one reason for the popularity of PDs is that these metrics are stable with respect to small deviations in the inputs (Cohen-Steiner et al., 2007, 2010; Chazal et al., 2014). Furthermore, the bottleneck metric allows one to define Fréchet means and variances for a collection of PDs (Mileyko et al., 2011; Turner et al., 2014). However, the structure of a metric space alone is insufficient for many ML techniques, and a recent area of interest in the topological data analysis community has been encoding PDs in ways that broaden the applicability of persistence. For example, Adcock et al. (2016) study a ring of algebraic functions on the space of persistence diagrams, and Verovšek (2016) identifies tropical coordinates on the space of diagrams. Ferri and Landi (1999) and Di Fabio and Ferri (2015) encode a PD using the coefficients of a complex polynomial that has the points of the PD as its roots.

Bubenik (2015) develops the notion of a persistence landscape, a stable functional representation of a PD that lies in a Banach space. A persistence landscape (PL) is a function $\lambda\colon \mathbb{N}\times\mathbb{R}\to[-\infty,\infty]$, which can equivalently be thought of as a sequence of functions $\lambda_k\colon \mathbb{R}\to[-\infty,\infty]$. For $1\leq p\leq\infty$ the p-landscape distance between two landscapes λ and λ' is defined as $\|\lambda-\lambda'\|_p$; the ∞-landscape distance is stable with respect to the bottleneck distance on PDs, and the p-landscape distance is continuous with respect to the p-Wasserstein distance on PDs. One of the motivations for defining persistence landscapes is that even though Fréchet means of PDs are not necessarily unique (Mileyko et al., 2011), a set of persistence landscapes does have a unique mean. Unique means are also a feature of PIs as they are vector representations. An advantage of PLs over PIs is that the map from a PD to a PL is easily invertible; an advantage of PIs over PLs is that PIs live in Euclidean space and hence are amenable to a broader range of ML techniques. In §6, we compare PDs, PLs, and PIs in a classification task on synthetic data sampled from common topological spaces. We find that PIs behave comparably or better than PDs when using ML techniques available to both representations, but PIs are significantly more efficient to compute. Also, PIs outperform PLs in the majority of the classification tasks and are of comparable computational efficiency.

A vector representation of a PD, due to Carrière et al. (2015), can be obtained by rearranging the entries of the distance matrix between points in a PD. In their Theorem 3.2, they prove that both the L^∞ and L^2 norms between their resulting vectors are stable with respect to the bottleneck distance on PDs. They remark that while the L^∞ norm is useful for nearest-neighbor classifiers, the L^2 norm allows for more elaborate algorithms such as SVM. However, though their stability result for the L^∞ norm is well-behaved, their constant for the L^2 norm scales undesirably with the number of points in the PD. We provide this as motivation for our Theorem 10, in which we prove the L^∞, L^1, and L^2 norms for PI vectors

are stable with respect to the 1-Wasserstein distance between PDs, and in which none of the constants depend on the number of points in the PD.

By superimposing a grid over a PD and counting the number of topological features in each bin, Rouse et al. (2015) create a feature vector representation. An advantage of this approach is that the output is easier to interpret than other more complicated representations, but a disadvantage is that the vectors are not stable for two reasons:

(i) an arbitrarily small movement of a point in a PD may move it to another bin, and

(ii) a PD point emerging from the diagonal creates a discontinuous change.

Source (i) of instability can be improved by first smoothing a PD into a surface. This idea has appeared multiple times in various forms—even prior to the development of persistent homology, Donatini et al. (1998) and Ferri et al. (1997) convert size functions (closely related to 0-dimensional PDs) into surfaces by taking a sum of Gaussians centered on each point in the diagram. This conversion is not stable due to (ii), and we view our work as a continued study of these surfaces, now also in higher homological dimensions, in which we introduce a weighting function[3] to address (ii) and obtain stability. Chung et al. (2009) produce a surface by convolving a PD with the characteristic function of a disk, and Pachauri et al. (2011) produce a surface by centering a Gaussian on each point, but both of these methods lack stability again due to (ii). Surfaces produced from random PDs are related to the empirical intensity plots of Edelsbrunner et al. (2012).

Reininghaus et al. (2015) produce a stable surface from a PD by taking the sum of a positive Gaussian centered on each PD point together with a negative Gaussian centered on its reflection below the diagonal; the resulting surface is zero along the diagonal. This approach is similar to PIs, and indeed we use a result of Reininghaus et al. (2015, Theorem 3) to show that persistence surfaces are stable only with respect to the 1-Wasserstein distance (Remark 6). Nevertheless, we propose our independently-developed surfaces as an alternative stable representation of PDs with the following potential advantages. First, the sum of non-negatively weighted Gaussians in PIs may be easier to interpret than a sum including negative Gaussians. Second, we produce vectors from persistence surfaces with well-behaved stability bounds, allowing one to use vector-based learning methods such as linear SVM. Indeed, Zeppelzauer et al. (2016) report that while the kernel of Reininghaus et al. (2015) can be used with nonlinear SVMs, in practice, this becomes inefficient for a large number of training vectors because the entire kernel matrix must be computed. Third, while the surface of Reininghaus et al. (2015) weights persistence points further from the diagonal more heavily, there are situations in which one may prefer different weightings, as discussed in §1 and item (v) of our Problem Statement. Hence, one may want weightings on PD points that are non-increasing or even decreasing when moving away from the diagonal, an option available in the PI approach.

We produce a persistence surface from a PD by taking a weighted sum of Gaussians centered at each point. We create vectors, or PIs, by integrating our surfaces over a grid, allowing ML techniques for finite-dimensional vector spaces to be applied to PDs. Persistence images are stable, and distinct homology dimensions may be concatenated together into a single vector to be analyzed simultaneously. Persistence surfaces are studied from the

3. Our weighting function is continuous and zero for points of zero persistence, i.e., points along the diagonal.

statistical point of view by Chen et al. (2015); their applications in Section 4 use the L^1 norm between these surfaces, which can be justified as a reasonable notion of distance due to Theorem 9 that proves the L^1 distance between such surfaces is stable.

Zeppelzauer et al. (2016) apply PIs to 3D surface analysis for archeological data, in which the machine learning task is to distinguish scans of natural rock surfaces from those containing ancient human-made engravings. The authors state they select PIs over other topological methods because PIs are computationally efficient and can be used with a broader set of ML techniques. PIs are compared to an aggregate topological descriptor for a PD: the first entry of this vector is the number of points in the diagram, and the remaining entries are the minimum, maximum, mean, standard deviation, variance, 1st-quartile, median, 3rd-quartile, sum of square roots, sum, and sum of squares of all the persistence values. In their three experiments, the authors find the following.

- When classifying natural rock surfaces from engravings using PDs produced from the sublevel set filtration, PIs outperform the aggregate descriptor.
- When the natural rock and engraved surfaces are first preprocessed using the completed local binary pattern (CLBP) operator for texture classificiation (Guo et al., 2010), PIs outperform the aggregate descriptor.
- The authors added PIs and the aggregate descriptor to eleven different non-topological baseline descriptors, and found that the classification accuracy of the baseline descriptor was improved more by the addition of PIs than by the addition of the aggregate descriptor.

Furthermore, Zeppelzauer et al. (2016, Table 1) demonstrate that for their machine learning task, PIs have low sensitivity to the parameter choices of resolution and variance (§4).

3. Background on Persistent Homology

Homology is an algebraic topological invariant that, roughly speaking, describes the holes in a space. The k-dimensional holes (connected components, loops, trapped volumes, etc.) of a topological space X are encoded in an algebraic structure called the k-th homology group of X, denoted $H_k(X)$. The rank of this group is referred to as the *k-th Betti number*, β_k, and counts the number of independent k-dimensional holes. For a comprehensive study of homology, see the textbook by Hatcher (2002).

Given a nested sequence of topological spaces $X_1 \subseteq X_2 \subseteq \ldots \subseteq X_n$, the inclusion $X_i \subseteq X_{i'}$ for $i \leq i'$ induces a linear map $H_k(X_i) \to H_k(X_{i'})$ on the corresponding k-th homology for all $k \geq 0$. The idea of *persistent homology* is to track elements of $H_k(X_i)$ as the scale (or "time") parameter i increases (Edelsbrunner and Harer, 2008; Zomorodian and Carlsson, 2005; Edelsbrunner and Harer, 2010). A standard way to represent persistent homology information is a *persistence diagram* (PD),[4] which is a multiset of points in the Cartesian plane $\mathbb{R}^2$. For a fixed choice of homological dimension k, each homological feature is represented by a point (x, y), whose *birth* and *death* indices x and y are the scale parameters at which that feature first appears and disappears, respectively. Since all topological

4. Another standard representation is the barcode (Ghrist, 2008).

features die after they are born, necessarily each point appears on or above the diagonal line $y = x$. A PD is a multiset of such points, as distinct topological features may have the same birth and death coordinates.[5] Points near the diagonal are often considered to be noise while those further from the diagonal represent more robust topological features.

In this paper, we produce PDs from two different types of input data:

(i) When the data is a point cloud, i.e., a finite set of points in some space, then we produce PDs using the Vietoris–Rips filtration.

(ii) When the data is a real-valued function, we produce PDs using the sublevel set filtration.[6]

For setting (i), point cloud data often comes equipped with a metric or a measure of internal (dis)similarity and is rich with latent geometric content. One approach to identifying geometric shapes in data is to consider the data set as the vertices of a simplicial complex and to add edges, triangles, tetrahedra, and higher-dimensional simplices whenever their diameter is less than a fixed choice of scale. This topological space is called the Vietoris–Rips simplicial complex, which we introduce in more detail in §A.2. The homology of the Vietoris–Rips complex depends crucially on the choice of scale, but persistent homology eliminates the need for this choice by computing homology over a range of scales (Carlsson, 2009; Ghrist, 2008). In §6.1–6.4.1, we obtain PDs from point cloud data using the Vietoris–Rips filtered simplicial complex, and we use ML techniques to classify the point clouds by their topological features.

In setting (ii), our input is a real valued function $f\colon X \to \mathbb{R}$ defined on some domain X. One way to understand the behavior of map f is to understand the topology of its sublevel sets $f^{-1}((-\infty, \epsilon])$. By letting ϵ increase, we obtain an increasing sequence of topological spaces, called the sublevel set filtration, which we introduce in more detail in §A.3. In §6.4.2, we obtain PDs from surfaces $u\colon [0,1]^2 \to \mathbb{R}$ produced from the Kuramoto-Sivashinsky equation, and we use ML techniques to perform parameter classification.

In both settings, the output of the persistent homology computation is a collection of PDs encoding homological features of the data across a range of scales. Let $\mathcal{D}$ denote the set of all PDs. The space $\mathcal{D}$ can be endowed with metrics as studied by Cohen-Steiner et al. (2007) and Mileyko et al. (2011). The *p-Wasserstein distance* defined between two PDs B and B' is given by

$$W_p(B, B') = \inf_{\gamma: B \to B'} \Big(\sum_{u \in B} ||u - \gamma(u)||_\infty^p \Big)^{1/p},$$

where $1 \leq p < \infty$ and γ ranges over bijections between B and B'. Another standard choice of distance between diagrams is $W_\infty(B, B') = \inf_{\gamma: B \to B'} \sup_{u \in B} ||u - \gamma(u)||_\infty$, referred to as the *bottleneck distance*. These metrics allow us to measure the (dis)similarity between the homological characteristics of two data sets.

5. By convention, all points on the diagonal are taken with infinite multiplicity. This facilitates the definitions of the p-Wasserstein and bottleneck distances below.
6. As explained in §A.3, (i) can be viewed as a special case of (ii).

4. Persistence Images

We propose a method for converting a PD into a vector while maintaining an interpretable connection to the original PD. Figure 1 illustrates the pipeline from data to PI starting with spectral and spatial information in $\mathbb{R}^5$ from an immunofluorescent image of a circulating tumor cell (Emerson et al., 2015).

Precisely, let B be a PD in birth-death coordinates.[7] Let $T\colon \mathbb{R}^2 \to \mathbb{R}^2$ be the linear transformation $T(x, y) = (x, y - x)$, and let $T(B)$ be the transformed multiset in birth-persistence coordinates,[8] where each point $(x, y) \in B$ corresponds to a point $(x, y - x) \in T(B)$. Let $\phi_u\colon \mathbb{R}^2 \to \mathbb{R}$ be a differentiable probability distribution with mean $u = (u_x, u_y) \in \mathbb{R}^2$. In all of our applications, we choose this distribution to be the normalized symmetric Gaussian $\phi_u = g_u$ with mean u and variance σ^2 defined as

$$g_u(x, y) = \frac{1}{2\pi\sigma^2} e^{-[(x-u_x)^2+(y-u_y)^2]/2\sigma^2}.$$

We fix a nonnegative weighting function $f\colon \mathbb{R}^2 \to \mathbb{R}$ that is zero along the horizontal axis, continuous, and piecewise differentiable. With these ingredients, we transform the PD into a scalar function over the plane.

Definition 1 *For B a PD, the corresponding* persistence surface $\rho_B\colon \mathbb{R}^2 \to \mathbb{R}$ *is the function*

$$\rho_B(z) = \sum_{u \in T(B)} f(u)\phi_u(z).$$

The weighting function f is critical to ensure the transformation from a PD to a persistence surface is stable, which we prove in §5.

Finally, the surface $\rho_B(z)$ is reduced to a finite-dimensional vector by discretizing a relevant subdomain and integrating $\rho_B(z)$ over each region in the discretization. In particular, we fix a grid in the plane with n boxes (pixels) and assign to each the integral of ρ_B over that region.

Definition 2 *For B a PD, its* persistence image *is the collection of pixels $I(\rho_B)_p = \iint_p \rho_B \, dydx$.*

PIs provide a convenient way to combine PDs of different homological dimensions into a single object. Indeed, suppose in an experiment the PDs for H_0, H_1, …, H_k are computed. One can concatenate the PI vectors for H_0, H_1, …, H_k into a single vector representing all homological dimensions simultaneously, and then use this concatenated vector as input into ML algorithms.

When generating a PI, the user makes three choices: the resolution, the distribution (and its associated parameters), and the weighting function. A strength of PIs is that they are flexible; a weakness is that these choices are noncanonical.

7. We omit points that correspond to features with infinite persistence, e.g., the H_0 feature corresponding to the connectedness of the complete simplicial complex.
8. Instead of birth-persistence coordinates, one could also use other choices such as birth-death or (average size)-persistence coordinates. Our stability results (§5) still hold with only a slight modification to the constants.

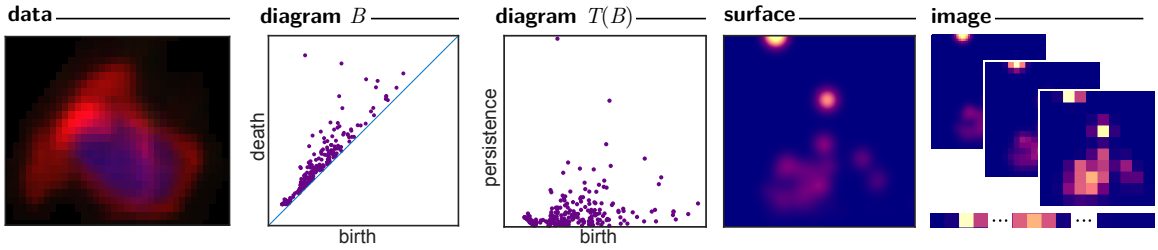

Figure 1: Algorithm pipeline to transform data into a persistence image.

Resolution of the image: The resolution of the PI corresponds to the grid being overlaid on the PD. The classification accuracy in the PI framework appears to be fairly robust to choice of resolution, as discussed in §6.2 and by Zeppelzauer et al. (2016).

The Distribution: Our method requires the choice of a probability distribution associated to each point in the PD. The examples in this paper use a Gaussian centered at each point, but other distributions may be used. The Gaussian distribution depends on a choice of variance: we leave this choice as an open problem, though the experiments in §6.2 and those of Zeppelzauer et al. (2016) show a low sensitivity to the choice of variance.

The Weighting Function: In order for our stability results in §5 to hold, our weighting function $f\colon \mathbb{R}^2 \to \mathbb{R}$ must be zero along the horizontal axis (the analogue of the diagonal in birth-persistence coordinates), continuous, and piecewise differentiable. A simple choice is a weighting function that depends only on the vertical persistence coordinate y. In order to weight points of higher persistence more heavily, functions which are nondecreasing in y, such as sigmoidal functions, are a natural choice. However, in certain ML tasks such as the work of Bendich et al. (2016) the points of small or medium persistence may perform best, and hence one may choose to use more general weighting functions. In our experiments in §6, we use a piecewise linear weighting function $f\colon \mathbb{R}^2 \to \mathbb{R}$ which only depends on the persistence coordinate y. Given $b > 0$, define $w_b\colon \mathbb{R} \to \mathbb{R}$ via

$$w_b(t) = \begin{cases} 0 & \text{if } t \leq 0, \\ \frac{t}{b} & \text{if } 0 < t < b, \text{ and} \\ 1 & \text{if } t \geq b. \end{cases}$$

We use $f(x, y) = w_b(y)$, where b is the persistence value of the most persistent feature in all trials of the experiment.

In the event that the birth coordinate is zero for all points in the PD, as is often the case for H_0, it is possible to generate a 1-dimensional (instead of 2-dimensional) PI using 1-dimensional distributions. This is the approach we adopt. Appendix B displays examples of PIs for the common topological spaces of a circle and a torus with various parameter choices.

5. Stability of Persistence Surfaces and Images

Due to the unavoidable presence of noise or measurement error, tools for data analysis ought to be stable with respect to small perturbations of the inputs. Indeed, one reason for the popularity of PDs in topological data analysis is that the transformation of a data

set to a PD is stable (Lipschitz) with respect to the bottleneck metric and—given some mild assumptions about the underlying data—is also stable with respect to the Wasserstein metrics (Edelsbrunner and Harer (2010)). In §5.1, we show that persistence surfaces and images are stable with respect to the 1-Wasserstein distance between PDs. In §5.2, we prove stability with improved constants when the PI is constructed using the Gaussian distribution.

5.1 Stability for general distributions

For $h\colon \mathbb{R}^2 \to \mathbb{R}$ differentiable, define $|\nabla h| = \sup_{z\in\mathbb{R}^2} \|\nabla h(z)\|_2$ to be the maximal norm of the gradient vector of h, i.e., the largest directional derivative of h. It follows by the fundamental theorem of calculus for line integrals that for all $u, v \in \mathbb{R}^2$, we have

$$|h(u) - h(v)| \leq |\nabla h|\ \|u - v\|_2. \tag{1}$$

Recall $\phi_u\colon \mathbb{R}^2 \to \mathbb{R}$ is a differentiable probability distribution with mean $u = (u_x, u_y) \in \mathbb{R}^2$. We may safely denote $|\nabla\phi_u|$ by $|\nabla\phi|$ and $\|\phi_u\|_\infty$ by $\|\phi\|_\infty$ since the maximal directional derivative and supremum of a fixed differentiable probability distribution are invariant under translation. Note that

$$\|\phi_u - \phi_v\|_\infty \leq |\nabla\phi|\ \|u - v\|_2 \tag{2}$$

since for any $z \in \mathbb{R}^2$ we have $|\phi_u(z) - \phi_v(z)| = |\phi_u(z) - \phi_u(z + u - v)| \leq |\nabla\phi|\ \|u - v\|_2$.

Recall that our nonnegative weighting function $f\colon \mathbb{R}^2 \to \mathbb{R}$ is defined to be zero along the horizontal axis, continuous, and piecewise differentiable.

Lemma 3 *For $u, v \in \mathbb{R}^2$, we have $\|f(u)\phi_u - f(v)\phi_v\|_\infty \leq \big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)\|u - v\|_2$.*

Proof For any $z \in \mathbb{R}^2$, we have

$$\begin{aligned}
|f(u)\phi_u(z) - f(v)\phi_v(z)| &= \big|f(u)\big(\phi_u(z) - \phi_v(z)\big) + \big(f(u) - f(v)\big)\phi_v(z)\big| \\
&\leq \|f\|_\infty\ |\phi_u(z) - \phi_v(z)| + \|\phi\|_\infty|f(u) - f(v)| \\
&\leq \|f\|_\infty|\nabla\phi|\ \|u - v\|_2 + \|\phi\|_\infty|\nabla f|\ \|u - v\|_2 \qquad \text{by (2) and (1)} \\
&= \big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)\|u - v\|_2.
\end{aligned}$$

■

Theorem 4 *The persistence surface ρ is stable with respect to the 1-Wasserstein distance between diagrams: for $B, B' \in \mathcal{D}$ we have*

$$\|\rho_B - \rho_{B'}\|_\infty \leq \sqrt{10}\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)W_1(B, B').$$

Proof Since we assume B and B' consist of finitely many points, there exists a matching γ that achieves the infimum in the Wasserstein distance. Then

$$\begin{aligned}
\|\rho_B - \rho_{B'}\|_\infty &= \| \sum_{u\in T(B)} f(u)\phi_u - \sum_{u\in T(B)} f(\gamma(u))\phi_{\gamma(u)}\|_\infty \\
&\leq \sum_{u\in T(B)} \|f(u)\phi_u - f(\gamma(u))\phi_{\gamma(u)}\|_\infty \\
&\leq \big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big) \sum_{u\in T(B)} \|u-\gamma(u)\|_2 \qquad \text{by Lemma 3.} \\
&\leq \sqrt{2}\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big) \sum_{u\in T(B)} \|u-\gamma(u)\|_\infty \quad \text{since } \|\cdot\|_2 \leq \sqrt{2}\|\cdot\|_\infty \text{ in } \mathbb{R}^2 \\
&\leq \sqrt{10}\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big) \sum_{u\in B} \|u-\gamma(u)\|_\infty \quad \text{since } \|T(\cdot)\|_2 \leq \sqrt{5}\|\cdot\|_\infty \\
&= \sqrt{10}\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big) W_1(B,B').
\end{aligned}$$

The step transforming from a sum over all $u \in T(B)$ to one over all $u \in B$ is necessary because the Wasserstein distance is defined using birth-death coordinates, not birth-persistence coordinates. The bound $\|T(\cdot)\|_2 \leq \sqrt{5}\|\cdot\|_\infty$ follows from the fact the unit ball with respect to the L^∞ norm in $\mathbb{R}^2$ (i.e., a square) gets mapped under T to a parallelogram contained inside a ball with respect to the L^2 norm of radius $\sqrt{5}$. ■

It follows that persistence images are also stable.

Theorem 5 *The persistence image $I(\rho_B)$ is stable with respect to the 1-Wasserstein distance between diagrams. More precisely, if A is the maximum area of any pixel in the image, A' is the total area of the image, and n is the number of pixels in the image, then*

$$\begin{aligned}
\|I(\rho_B) - I(\rho_{B'})\|_\infty &\leq \sqrt{10}A\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)W_1(B,B') \\
\|I(\rho_B) - I(\rho_{B'})\|_1 &\leq \sqrt{10}A'\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)W_1(B,B') \\
\|I(\rho_B) - I(\rho_{B'})\|_2 &\leq \sqrt{10n}A\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)W_1(B,B').
\end{aligned}$$

The constant for the L^2 norm bound containing $\sqrt{n}$ goes to infinity as the resolution of the image increases. For this reason, in Theorem 10 we provide bounds with better constants in the specific case of Gaussian distributions.

Proof Note for any pixel p with area $A(p)$ we have

$$\begin{aligned}
|I(\rho_B)_p - I(\rho_{B'})_p| &= \Big|\iint_p \rho_B \; dydz - \iint_p \rho_{B'} \; dydx\Big| \\
&= \Big|\iint_p \rho_B - \rho_{B'} \; dydx\Big| \\
&\leq A(p)\|\rho_B - \rho_{B'}\|_\infty \\
&\leq \sqrt{10}A(p)\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)W_1(B,B') \qquad \text{by Theorem 4.}
\end{aligned}$$

Hence we have

$$\begin{aligned}\|I(\rho_B) - I(\rho_{B'})\|_\infty &\leq \sqrt{10}A\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)W_1(B,B')\\ \|I(\rho_B) - I(\rho_{B'})\|_1 &\leq \sqrt{10}A'\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)W_1(B,B')\\ \|I(\rho_B) - I(\rho_{B'})\|_2 &\leq \sqrt{n}\|I(\rho_B) - I(\rho_{B'})\|_\infty\\ &\leq \sqrt{10n}A\big(\|f\|_\infty|\nabla\phi| + \|\phi\|_\infty|\nabla f|\big)W_1(B,B').\end{aligned}$$

■

Remark 6 *Recall $\mathcal{D}$ is the set of all PDs. The kernel $k\colon \mathcal{D}\times\mathcal{D}\to\mathbb{R}$ defined by $k(B,B') = \langle I(\rho_B), I(\rho_{B'})\rangle_{\mathbb{R}^n}$ is non-trivial and additive, and hence Theorem 3 of Reininghaus et al. (2015) implies that k is not stable with respect to W_p for any $1 < p \leq \infty$. That is, when $1 < p \leq \infty$ there is no constant c such that for all $B, B' \in \mathcal{D}$ we have $\|I(\rho_B) - I(\rho_{B'})\|_2 \leq cW_p(B,B')$.*

5.2 Stability for Gaussian distributions

In this section, we provide stability results with better constants in the case of Gaussian distributions. With Gaussian distributions, we can control not only the L^∞ distance but also the L^1 distance between two persistence surfaces.

Our results for 2-dimensional Gaussians will rely on the following lemma for 1-dimensional Gaussians.

Lemma 7 *For $u, v \in \mathbb{R}$, let $g_u, g_v\colon \mathbb{R}\to\mathbb{R}$ be the normalized 1-dimensional Gaussians, defined via $g_u(z) = \dfrac{1}{\sigma\sqrt{2\pi}}e^{-(z-u)^2/2\sigma^2}$. If $a, b > 0$, then*

$$\|ag_u - bg_v\|_1 \leq |a-b| + \sqrt{\frac{2}{\pi}}\frac{\min\{a,b\}}{\sigma}|u-v|.$$

Proof Let $\mathrm{Erf}(t) = \frac{2}{\sqrt{\pi}}\int_0^t e^{-u^2}du$. We show in Appendix C that

$$\|ag_u - bg_v\|_1 = F(v-u), \tag{3}$$

where $F\colon\mathbb{R}\to\mathbb{R}$ is defined by

$$F(z) = \begin{cases} |a-b| & \text{if } z = 0\\ \left|a\mathrm{Erf}\left(\frac{z^2+2\sigma^2\ln(a/b)}{z\sigma2\sqrt{2}}\right) - b\mathrm{Erf}\left(\frac{-z^2+2\sigma^2\ln(a/b)}{z\sigma2\sqrt{2}}\right)\right| & \text{otherwise.}\end{cases}$$

The roots of F'' are $z = \pm\sigma\sqrt{2\ln(a/b)}$ and $z = \pm i\sigma\sqrt{2\ln(a/b)}$. If $a > b$, the unique positive real root is $z_a \equiv \sigma\sqrt{2\ln(a/b)}$ while if $b > a$, the unique positive real root is $z_b \equiv -i\sigma\sqrt{2\ln(a/b)}$. Since $F'(z_a) = b\sqrt{2/\pi}/\sigma$ and $F'(z_b) = a\sqrt{2/\pi}/\sigma$, we conclude that

$$\|F'\|_\infty = \sqrt{\frac{2}{\pi}}\frac{\min\{a,b\}}{\sigma}, \quad \text{and hence} \quad F(z) \leq |a-b| + \sqrt{\frac{2}{\pi}}\frac{\min\{a,b\}}{\sigma}|z|.$$

The result follows by letting $z = v - u$. ■

Lemma 8 *For $u, v \in \mathbb{R}^2$, let $g_u, g_v \colon \mathbb{R}^2 \to \mathbb{R}$ be normalized 2-dimensional Gaussians. Then*

$$\|f(u)g_u - f(v)g_v\|_1 \leq \left(|\nabla f| + \sqrt{\frac{2}{\pi}} \frac{\min\{f(u), f(v)\}}{\sigma} \right) \|u - v\|_2.$$

The proof of Lemma 8 is shown in Appendix C and uses a similar construction to that of Lemma 7. We are prepared to prove the stability of persistence surfaces with Gaussian distributions.

Theorem 9 *The persistence surface ρ with Gaussian distributions is stable with respect to the 1-Wasserstein distance between diagrams: for $B, B' \in \mathcal{D}$ we have*

$$\|\rho_B - \rho_{B'}\|_1 \leq \left(\sqrt{5}|\nabla f| + \sqrt{\frac{10}{\pi}} \frac{\|f\|_\infty}{\sigma} \right) W_1(B, B').$$

Proof Since we assume B and B' consist of finitely many off-diagonal points, there exists a matching γ that achieves the infimum in the Wasserstein distance. Then

$$\begin{aligned}
\|\rho_B - \rho_{B'}\|_1 &= \left\| \sum_{u \in T(B)} f(u) g_u - \sum_{u \in T(B)} f(\gamma(u)) g_{\gamma(u)} \right\|_1 \\
&\leq \sum_{u \in T(B)} \|f(u)g_u - f(\gamma(u))g_{\gamma(u)}\|_1 \\
&\leq \left(|\nabla f| + \sqrt{\frac{2}{\pi}} \frac{\|f\|_\infty}{\sigma} \right) \sum_{u \in T(B)} \|u - \gamma(u)\|_2 \quad \text{by Lemma 8, where } \min\{f(u), f(v)\} \leq \|f\|_\infty \\
&\leq \left(\sqrt{5}|\nabla f| + \sqrt{\frac{10}{\pi}} \frac{\|f\|_\infty}{\sigma} \right) \sum_{u \in B} \|u - \gamma(u)\|_\infty \quad \text{since } \|T(\cdot)\|_2 \leq \sqrt{5}\|\cdot\|_\infty \\
&= \left(\sqrt{5}|\nabla f| + \sqrt{\frac{10}{\pi}} \frac{\|f\|_\infty}{\sigma} \right) W_1(B, B').
\end{aligned}$$

■

It follows that persistence images are also stable.

Theorem 10 *The persistence image $I(\rho_B)$ with Gaussian distributions is stable with respect to the 1-Wasserstein distance between diagrams. More precisely,*

$$\|I(\rho_B) - I(\rho_{B'})\|_1 \leq \left(\sqrt{5}|\nabla f| + \sqrt{\frac{10}{\pi}} \frac{\|f\|_\infty}{\sigma} \right) W_1(B, B')$$

$$\|I(\rho_B) - I(\rho_{B'})\|_2 \leq \left(\sqrt{5}|\nabla f| + \sqrt{\frac{10}{\pi}} \frac{\|f\|_\infty}{\sigma} \right) W_1(B, B')$$

$$\|I(\rho_B) - I(\rho_{B'})\|_\infty \leq \left(\sqrt{5}|\nabla f| + \sqrt{\frac{10}{\pi}} \frac{\|f\|_\infty}{\sigma} \right) W_1(B, B').$$

Proof We have

$$\|I(\rho_B) - I(\rho_{B'})\|_1 = \sum_p \left| \iint_p \rho_B \, dydz - \iint_p \rho_{B'} \, dydx \right| \leq \iint_{\mathbb{R}^2} |\rho_B - \rho_{B'}| \, dydz$$
$$= \|\rho_B - \rho_{B'}\|_1 \leq \left(\sqrt{5} |\nabla f| + \sqrt{\frac{10}{\pi}} \frac{\|f\|_\infty}{\sigma} \right) W_1(B, B')$$

by Theorem 9. The claim follows since $\|\cdot\|_2 \leq \|\cdot\|_1$ and $\|\cdot\|_\infty \leq \|\cdot\|_1$ for vectors in $\mathbb{R}^n$. ■

6. Experiments

We perform several experiments in order to assess the added value of our vector representation of PDs. First, in §6.1, we compare the performance of PDs, PLs, and PIs in a classification task for a synthetic data set consisting of point clouds sampled from six different topological spaces using K-medoids, which requires only a metric space (instead of a vector space) structure. We find that PIs produce consistently high classification accuracy, and furthermore, the computation time for PIs is significantly faster than computing bottleneck or Wasserstein distances between PDs. In §6.2, we explore the impact that the choices of parameters determining our PIs have on classification accuracy. We find that the accuracy is insensitive to the particular choices of PI resolution and distribution variance. In §6.3, we combine PIs with a sparse support vector machine classifier to identify the most strongly differentiating pixels for classification; this is an example of a ML task which is facilitated by the fact that PIs are finite vectors. Finally, as a novel machine learning application, we illustrate the utility of PIs to infer dynamical parameter values in both continuous and discrete dynamical systems: a discrete time system called the linked twist map in §6.4.1, and a partial differential equation called the anisotropic Kuramoto-Sivashinsky equation in §6.4.2.

6.1 Comparison of PDs, PLs, and PIs using K-medoids Classification

Our synthetic data set consists of six shape classes: a unit cube, a circle of diameter one, a sphere of diameter one, three clusters with centers randomly chosen in the unit cube, three clusters within three clusters (where the centers of the minor clusters are chosen as small—i.e., <0.1—random perturbations from the major cluster centers), and a torus with a major diameter of one and a minor diameter of one half. We produce 25 point clouds of 500 points sampled uniformly at random from each of the six shapes, and then add a level of Gaussian noise. This gives 150 point clouds in total.

We then compute the H_0 and H_1 PDs for the Vietoris–Rips filtration (§A.2) built from each point cloud which have been endowed with the ambient Euclidean metric on $\mathbb{R}^3$.

Our goal is to compare various methods for transforming PDs into distance matrices to be used to establish proximity of topological features extracted from data. We create $3^2 \cdot 2^2 = 36$ distance matrices of size 150×150, using three choices of representation (PDs,

PLs, PIs), three choices of metric (L^1, L^2, L^∞),[9] two choices of Gaussian noise ($\eta = 0.05$, 0.1), and two homological dimensions (H_0, H_1). For example, the PD, H_1, L^2, $\eta = 0.1$, distance matrix contains the 2-Wasserstein distances between the H_1 PDs for the random point clouds with noise level 0.1. By contrast, the PI, H_1, L^2, $\eta = 0.1$ distance matrix contains all pairwise L^2 distances between the PIs[10] produced from the H_1 PDs with noise level 0.1.

We first compare these distance matrices based on how well they classify the random point clouds into shape classes via K-medoids clustering (Kaufman and Rousseeuw, 1987; Park and Jun, 2009). K-medoids produces a partition of a metric space into K clusters by choosing K points from the data set called *medoids* and assigning each metric space point to its closest medoid. The *score* of such a clustering is the sum of the distances from each point to its closest medoid. The desired output of K-medoids is the clustering with the minimal clustering score. Unfortunately, an exhaustive search for the global minimum is often prohibitively expensive. A typical approach to search for this global minimum is to choose a large selection of K random initial medoids, improve each selection of medoids iteratively in rounds until the clustering score stabilizes and then return the identified final clustering with the lowest score for each initialization. In our experiments, we choose 1,000 random initial selections of $K = 6$ medoids (as there are six shape classes) for each distance matrix, improve each selection of medoids using the Voronoi iteration method (Park and Jun, 2009), and return the clustering with the lowest classification score. To each K-medoids clustering we assign an accuracy which is equal to the percentage of random point clouds identifed with a medoid of the same shape class. In Table 1, we report the classification accuracy of the K-medoids clustering with the lowest clustering score, for each distance matrix.

Our second criterion for comparing methods to produce distance matrices is computational efficiency. In Table 1, we report the time required to produce each distance matrix, starting with 150 precomputed PDs as input. In the case of PLs and PIs, this time includes the intermediate step of transforming each PD into the alternate representation, as well as computing the pairwise distance matrix. All timings are computed on a laptop with a 1.3 GHz Intel Core i5 processor and 4 GB of memory. We compute bottleneck, 1-Wasserstein, and 2-Wasserstein distance matrices using the software of Kerber et al. (2016). For PL computations, we use the Persistence Landscapes Toolbox by Bubenik and Dlotko (2016). Our MATLAB code for producing PIs is publically available at `https://github.com/CSU-TDA/PersistenceImages`.

We see in Table 1 that PI distance matrices have higher classification accuracy than nearly every PL distance matrix, and higher classification accuracy than PDs in half of the trials.

Furthermore, the computation times for PI distance matrices are significantly lower than the time required to produce distance matrices from PDs using the bottleneck or p-Wasserstein metrics. There is certainly no guarantee that PIs will outperform PDs or PLs in any given machine learning task. However, in this experiment, persistent images provide

9. The L^1, L^2, L^∞ distances on PDs are more commonly known as the 1-Wasserstein, 2-Wasserstein, and bottleneck distances.

10. For PIs in this experiment, we use variance $\sigma = 0.1$, resolution 20×20, and the weighting function defined in §4.

Table 1: Comparing classification accuracy and times of PDs, PLs, and PIs. The timings contain the computation time in seconds for producing a 150×150 distance matrix from 150 precomputed PDs. In the case of PLs and PIs, this requires first transforming each PD into its alternate representation and then computing a distance matrix. We consider 36 distinct distance matrices: three representations (PDs, PLs, PIs), two homological dimensions (H_0, H_1), three choices of metric (L^1, L^2, L^∞), and two levels of Gaussian noise ($\eta = 0.05$, 0.1).

Distance Matrix	Accuracy (Noise 0.05)	Time (Noise 0.05)	Accuracy (Noise 0.1)	Time (Noise 0.1)
PD, H_0, L^1	96.0%	37346s	96.0%	42613s
PD, H_0, L^2	91.3%	24656s	91.3%	25138s
PD, H_0, L^∞	60.7%	1133s	63.3%	1149s
PD, H_1, L^1	100%	657s	96.0%	703s
PD, H_1, L^2	100%	984s	97.3%	1042s
PD, H_1, L^∞	81.3%	527s	66.7%	564s
PL, H_0, L^1	92.7%	29s	96.7%	33s
PL, H_0, L^2	77.3%	29s	82.0%	34s
PL, H_0, L^∞	60.7%	2s	63.3%	2s
PL, H_1, L^1	83.3%	36s	80.7%	48s
PL, H_1, L^2	83.3%	50s	66.7%	69s
PL, H_1, L^∞	74.7%	8s	66.7%	9s
PI, H_0, L^1	93.3%	9s	95.3%	9s
PI, H_0, L^2	92.7%	9s	95.3%	9s
PI, H_0, L^∞	94.0%	9s	96.0%	9s
PI, H_1, L^1	100%	17s	95.3%	18s
PI, H_1, L^2	100%	17s	96.0%	18s
PI, H_1, L^∞	100%	17s	96.0%	18s

a representation of persistent diagrams which is both useful for the classification task and also computationally efficient.

6.2 Effect of PI Parameter Choice

In any system that relies on multiple parameters, it is important to understand the effect of parameter values on the system. As such, we complete a search of the parameter space used to generate PIs on the shape data set described in §6.1 and measure K-medoids classification accuracy as a function of the parameters. We use 20 different resolutions (ranging from 5×5 to 100×100 in increments of 5), a Gaussian function with 20 different choices of variance (ranging from 0.01 to 0.2 in increments of 0.01), and the weighting function described in §4. For each set of parameters, we compute the classification accuracy of the K-medoids

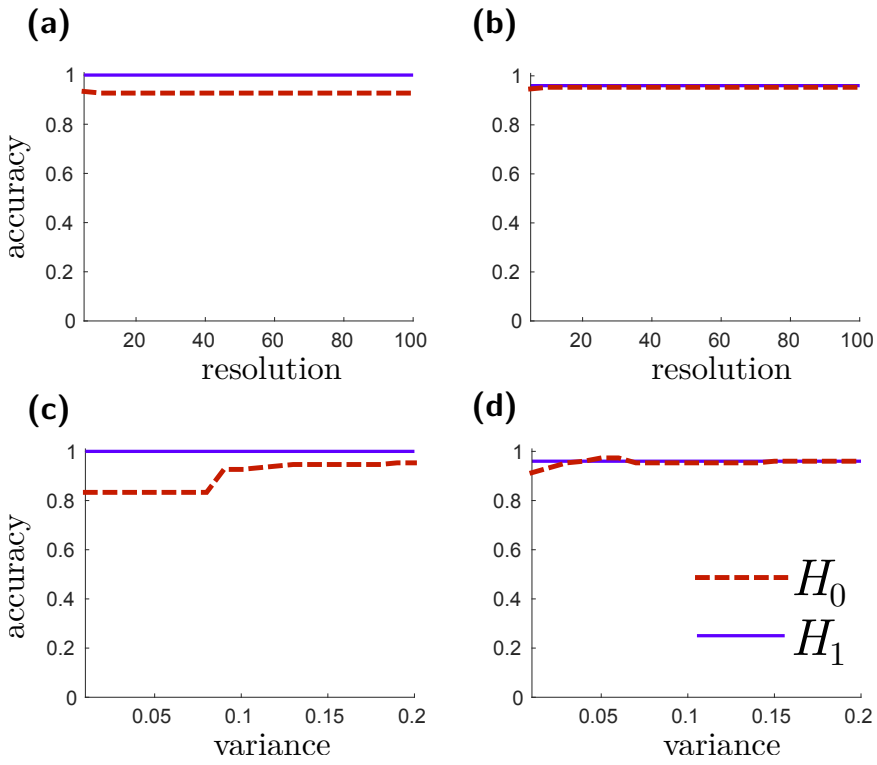

Figure 2: K-medoids classification accuracy as a function of resolution and variance for the data set of six shape classes. First column: noise level $\eta = 0.05$. Second column: noise level $\eta = 0.1$. First row: fixed variance 0.1 with resolutions ranging from 5×5 to 100×100 in increments of 5. Second row: fixed resolution 20×20 with variances ranging from 0.01 to 0.2 in increments of 0.01.

clustering with the minimum clustering score on the two sets of noise levels for the homology dimensions H_0 and H_1. We observe that the classification accuracy is insensitive to the choice of resolution and variance.

The plots in Figure 2 are characteristic of the 2-dimensional accuracy surface over all combinations of parameters in the ranges of variances and resolutions we tested. In an application to archeology, Zeppelzauer et al. (2016) find a similar robustness of PIs to the choices of resolution and variance.

6.3 Differentiating Homological Features by Sparse Support Vector Machine

The 1-norm regularized *linear* support vector machine (SVM), a.k.a. sparse SVM (SSVM) classifies data by generating a separating hyperplane that depends on very few input space features (Bradley and Mangasarian, 1998; Zhu et al., 2004; Zhang and Zhou, 2010). Such a model can be used for reducing data dimension or selecting discriminatory features. Note that linear SSVM feature selection is implemented on vectors and therefore, can be used on our PIs to select discriminatory pixels during classification. Other PD representations in the literature (Reininghaus et al., 2015; Pachauri et al., 2011) are designed to use kernel ML methods, such as *kernel* (nonlinear) SVMs. However, constructing kernel SVM classifiers

using the 1-norm results in minimizing the number of kernel functions, not the number of features in the input space (i.e., pixels in our application) (Fung and Mangasarian, 2004). Hence, for the purpose of feature selection or more precisely, PI pixel selection, we employ the linear SSVM.

We adopt the one-against-all (OAA) SSVM on the sets of H_0 and H_1 PIs from the six class shape data. In a one-against-all SSVM, there is one binary SSVM for each class to separate members of that class from members of all other classes. The PIs were generated using resolution 20×20, variance 0.0001, and noise level 0.05. Note that because of the resolution parameter choice of 20×20, each PI is a 400-dimensional vector, and the selected features will be a subset of indices corresponding to pixels within the PI. Using 5-fold cross-validated SSVM resulted in 100% accuracy comparing six sparse models with indications of the discriminatory features. Feature selection is performed by retaining the features (again, in this application, pixels) with non-zero SSVM weights, determined by magnitude comparison using weight ratios; for details see Chepushtanova et al. (2014). Figure 3 provides two examples, indicating the pixels of H_1 PIs that discriminate circles and tori from the other classes in the synthetic data set.

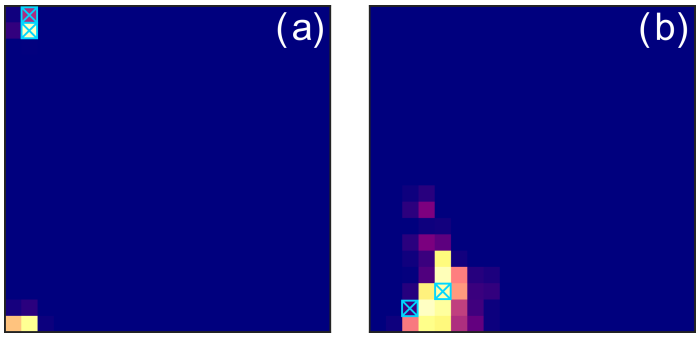

Figure 3: SSVM-based feature (pixel) selection for H_1 PIs from two classes of the synthetic data. Selected pixels are marked by blue crosses. (a) A noisy circle with the two selected pixels (indices 21 and 22 out of 400). (b) A noisy torus with the two selected pixels (indices 59 and 98 out of 400). The PI parameters used are resolution 20×20 and variance 10^{-4}, for noise level 0.05.

Feature selection produces highly interpretable results. The discriminatory pixels in the H_1 PIs that separate circles from the other classes correspond to the region where highly persistent H_1 topological features exist across all samples of a noisy circle (highlighted in Figure 3a). Alternatively, the discriminatory pixels in H_1 PIs that separate tori from the other classes correspond to points of short to moderate persistence (see Figure 3b). In this way, Figure 3b reiterates an observation of Bendich et al. (2016) that points of short to moderate persistence can contain important discriminatory information. Similar conclusions can be drawn from the discriminatory pixels of others classes (Appendix D). Our classification accuracy of 100% is obtained using only those pixels selected by SSVM (a cumulative set of only 10 distinct pixels).

6.4 Application: Determination of Dynamical System Parameters

Models of dynamic physical phenomenon rarely agree perfectly with the reality they represent. This is often due to the presence of poorly-resolved (or poorly-understood) processes which are parameterized rather than treated explicitly. As such, determination of the influence of a model parameter—which may itself be an incompletely-described conglomeration of several physical parameters—on model dynamics is a mainstay of dynamical system analysis. In the case of fitting a dynamic model to data, i.e., explicit determination of optimal model parameters, a variety of techniques exist for searching through parameter space, which often necessitate costly simulations. Furthermore, such approaches struggle when applied to models exhibiting sensitivity to initial conditions. We recast this problem as a machine-learning exercise based on the hypotheses that model parameters will be reflected directly in dynamic data in a way made accessible by persistent homology.

6.4.1 A discrete dynamical model

We approach a classification problem with data arising from the linked twist map, a discrete dynamical system modeling fluid flow. Hertzsch et al. (2007) use the linked twist map to model flows in DNA microarrays with a particular interest in understanding turbulent mixing. This demonstrates a primary mechanism giving rise to chaotic advection. The linked twist map is a Poincaré section of *eggbeater-type flow* (Hertzsch et al., 2007) in continuous dynamical systems. The Poincaré section captures the behavior of the flow by viewing a particle's location at discrete time intervals. The linked twist map is given by the discrete dynamical system

$$\begin{aligned} x_{n+1} &= x_n + r y_n(1 - y_n) \mod 1 \\ y_{n+1} &= y_n + r x_n(1 - x_n) \mod 1, \end{aligned}$$

where r is a positive parameter. For some values of r, the orbits $\{(x_n, y_n) : n = 0, \ldots, \infty\}$ are dense in the domain. However, for other parameter values, voids form. In either case, the truncated orbits $\{(x_n, y_n) : n = 0, \ldots, N \in \mathbb{N}\}$ exhibit complex structure.

For this experiment, we choose a set of parameter values, $r = 2.5$, 3.5, 4.0, 4.1 and 4.3, which produce a variety of orbit patterns. For each parameter value, 50 randomly-chosen initial conditions are selected, and 1000 iterations of the linked twist map are used to generate point clouds in $\mathbb{R}^2$. Figure 4 shows examples of typical orbits generated for each parameter value. The goal is to classify the trials by parameter value using PIs to capitalize on distinguishing topological features of the data. We use resolution 20×20 and a Gaussian with variance $\sigma = 0.005$ to generate the PIs. These parameters were chosen after a preliminary parameter search and classification effort. Similar results hold for a range of PI parameter values.

For a fixed r parameter value and a large number of points (many thousands), the patterns in the distributions of iterates show only small visible variations for different choices of the initial condition (x_0, y_0). However, with few points, such as in Figure 5, there are more significant variations in the patterns for different choices of initial conditions, making classification more difficult.

We perform classification and cross-validation with a discriminant subspace ensemble. This ML algorithm trains many "weak" learners on randomly chosen subspaces of the data

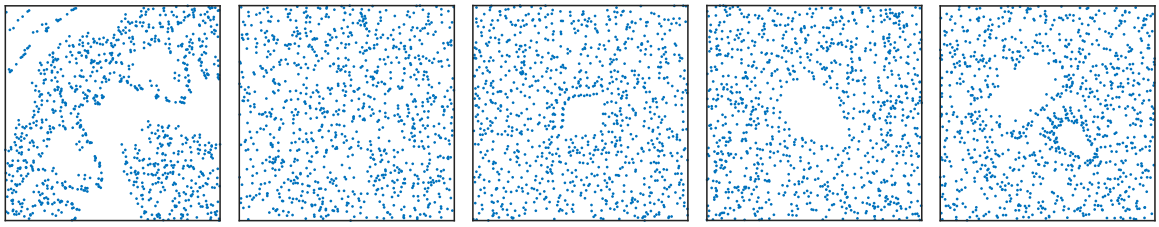

Figure 4: Examples of the first 1000 iterations, $\{(x_n, y_n) \ : \ n = 0, \ldots, 1000\}$, of the linked twist map with parameter values $r = 2$, 3.5, 4.0, 4.1 and 4.3, respectively.

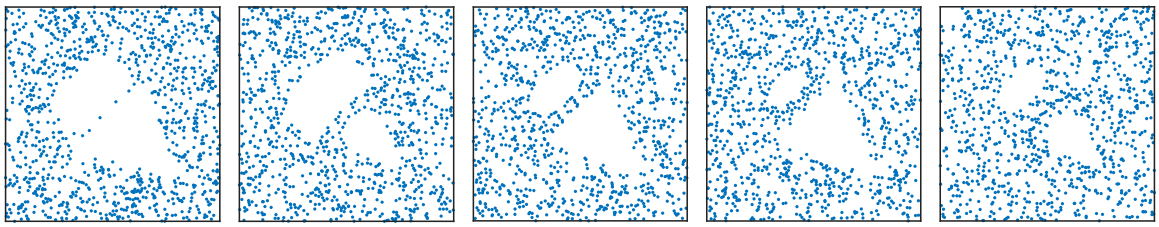

Figure 5: Truncated orbits, $\{(x_n, y_n) \ : \ n = 0, \ldots, 1000\}$, of the linked twist map with fixed $r = 4.3$ for different initial conditions (x_0, y_0).

(of a fixed dimension), and classifies and assigns a score to each point based on the current subspace. The final classification arises from an average of the scores of each data point over all learners (Ho, 1998). We perform 10 trials and average the classification accuracies. For the concatenated H_0 and H_1 PIs, this method achieves a classification accuracy of 82.5%; compared to 49.8% when using only H_0 PIs and 65.7% when using H_1 PIs. This experiment highlights two strengths of PIs: they offer flexibility in choosing a ML algorithm that is well suited to the data under consideration, and homological information from multiple dimensions may be leveraged simultaneously for greater discriminatory power.

This application is a brief example of the utility of PIs in classification of data from dynamical systems and modeling real-world phenomena, which provides a promising direction for further applications of PIs.

6.4.2 A partial differential equation

The Kuramoto-Sivashinsky (KS) equation is a partial differential equation for a function $u(x, y, t)$ of spatial variables x, y and time t that has been independently derived in a variety of problems involving pattern formation in extended systems driven far from equilibrium. Applications involving surface dynamics include surface nanopatterning by ion-beam erosion (Cuerno and Barabási, 1995; Motta et al., 2012), epitaxial growth (Villain, 1991; Wolf, 1991; Rost and Krug, 1995), and solidification from a melt (Golovin and Davis, 1998). In these applications, the nonlinear term in the KS equation may be anisotropic, resulting in the anisotropic Kuramoto-Sivashinsky (aKS) equation

$$\frac{\partial}{\partial t}u = -\nabla^2 u - \nabla^2\nabla^2 u + r\left(\frac{\partial}{\partial x}u\right)^2 + \left(\frac{\partial}{\partial y}u\right)^2, \tag{4}$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$, and the real parameter r controls the degree of anisotropy. At a fixed time t^*, $u(x, y, t^*)$ is a patterned surface (periodic in both x and y) defined over the (x, y)-plane. Visibly, the anisotropy appears as a slight tendency for the pattern to be elongated in the vertical or horizontal direction.

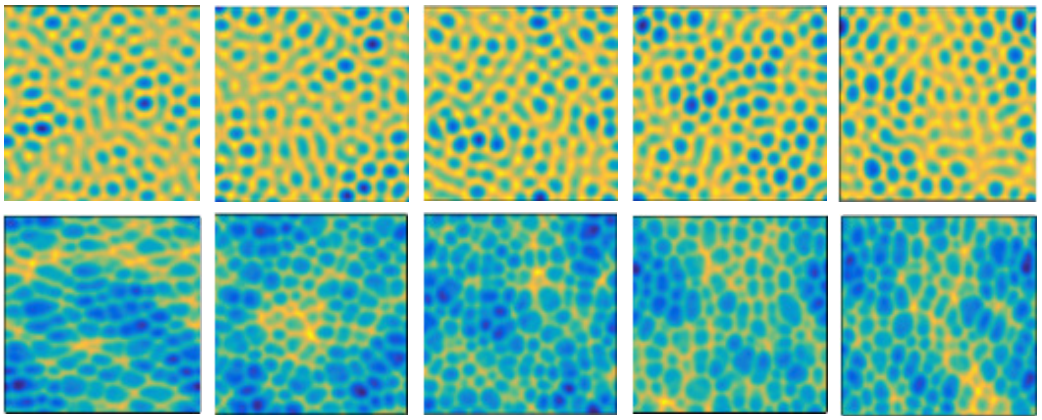

Figure 6: Plots of height-variance-normalized surfaces $u(x, y, \cdot)$ resulting from numerical simulations of the aKS equation (4). Each column represents a different parameter value: (from left) $r = 1$, 1.25, 1.5, 1.75 and 2. Each row represents a different time: $t = 3$ (top) and $t = 5$ (bottom). By $t = 5$ any anisotropic elongation of the surface pattern has visibly stabilized.

Numerical simulations of the aKS equation for a range of parameter values (columns) and simulation times (rows) are shown in Figure 6. For all simulations, the initial conditions were low-amplitude white noise. We employed a Fourier spectral method with periodic boundary conditions on a 512×512 spatial grid, with a fourth-order exponential time differencing Runge-Kutta method for the time stepping. Five values for the parameter r were chosen, namely $r = 1$, 1.25, 1.5, 1.75 and 2, and thirty trials were performed for each parameter value. Figure 7 shows the similarity between surfaces associated to two parameter values $r = 1.75$ and $r = 2$ at an early time.

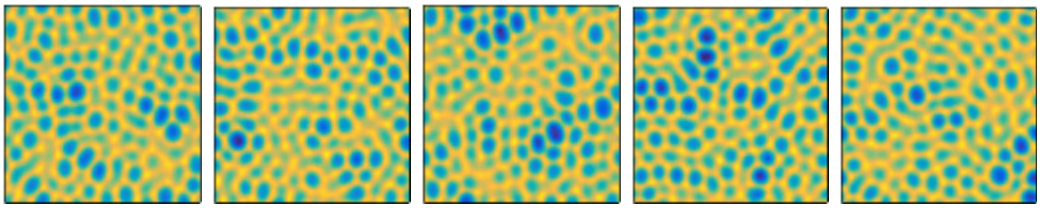

Figure 7: To illustrate the difficulty of our classification task, consider five instances of surfaces $u(x, y, 3)$ for $r = 1.75$ or $r = 2$, plotted on the same color axis. These surfaces are found by numerical integration of Equation (4), starting from random initial conditions. Can you group the images by eye?

Answer: (from left) $r = 1.75, 2, 1.75, 2, 2.$

We aim to identify the anisotropy parameter for each simulation using snapshots of surfaces $u(x, y, \cdot)$ as they evolve in time. Inference of the parameter using the surface alone proves difficult for several reasons. First, Equation (4) exhibits sensitivity to initial conditions: initially nearby solutions diverge quickly. Second, although the surface $u(x, y, t^*)$ at a fixed time is an approximation due to the finite discretization of its domain, the spatial resolution is still very large: in fact, these surfaces may be thought of as points in $\mathbb{R}^{266144}$. We were unable to perform standard classification techniques in this space. It was therefore necessary to perform some sort of dimension reduction. One such method is to simply 'resize' the surface by coarsening the discretization of the spatial domain after computing the simulation at a high resolution by replacing a block of grid elements with their average surface height. The surfaces were resized in this way to a resolution of 10×10 and a subspace discriminant ensemble was used to perform classification. Unsurprisingly, this method performs very poorly at all times (first row of Table 2).

The anisotropy parameter also influences the mean and amplitude of the surface pattern. We eliminate differences in the mean by mean-centering each surface after the simulation. To assess the impact of the variance of surface height on our task, classification was performed using a normal distribution-based classifier built on the variances of the surface heights. In this classifier, a normal distribution was fit to a training set of $2/3$ of the variances for each parameter value, and the testing data was classified based on a z-test for each of the different models. That is, a p-value for each new variance was computed for membership to the five normal distributions (corresponding to the five parameter choices of r), and the surface was classified based on the model yielding the highest p-value. After the pattern has more fully emerged (by, say, time $t = 5$) this method of classification yields 75% accuracy,[11] as shown in Table 2. However, early on in the formation of the pattern, this classifier performs very poorly because height variance is not yet a discriminating feature. Figure 8 shows the normal distribution fit to the variance of the surfaces for each parameter value at times $t = 3$ and 5, and illustrates why the variance of surface height is informative only after a surface is allowed to evolve for a sufficiently long time.

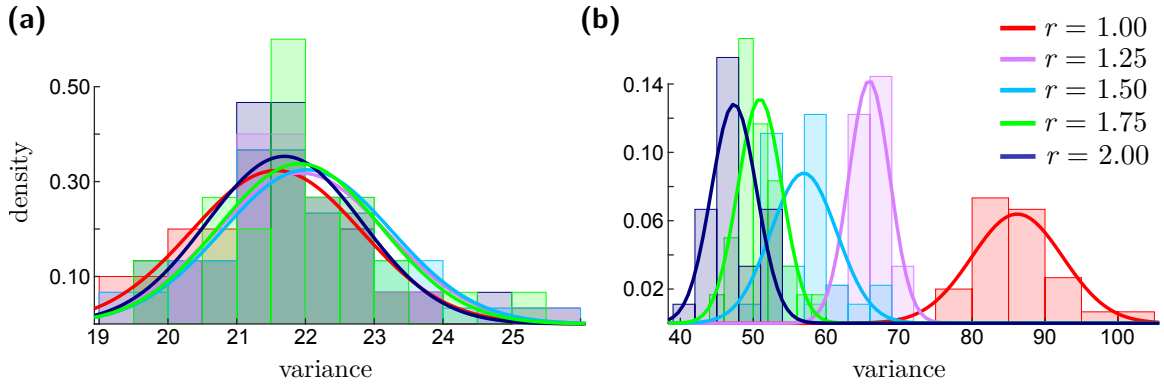

Figure 8: Histograms of the variances of surface heights for each parameter value, and the normal distribution fit to each histogram, for times (a) $t = 3$ and (b) $t = 5$.

11. Accuracy reported is averaged over 100 different training and testing partitions.

Table 2: Classification accuracies at different times of the aKS solution, using different classification approaches. Classification of times $t = 15$ and 20 result in accuracies similar to t =10.

Classification Approach	Time t=3	Time t=5	Time t=10
Subspace Discriminant Ensemble, Resized Surfaces	26.0 %	19.3%	19.3 %
Variance Normal Distribution Classifier	20.74%	75.2%	77.62 %
Subspace Discriminant Ensemble, H_0 PIs	58.3 %	96.0 %	94.7 %
Subspace Discriminant Ensemble, H_1 PIs	67.7 %	87.3 %	93.3%
Subspace Discriminant Ensemble, H_0 and H_1 PIs	72.7 %	95.3 %	97.3 %

Variance of a surface is reflected in its sublevel set filtration (see §A.3 for more details) PD. Yet, the PD and the subsequent PI contain additional topological structure, which may reveal other influences of the anisotropy parameter on the evolution of the surface. Persistence diagrams were computed using the sublevel set filtration, and PIs were generated with resolution 10×10 and a Gaussian with variance $\sigma = 0.01$. We think of our pipeline to a PI as a dimensionality reduction in this case, taking a surface which in actuality is a very high-dimensional point and producing a much lower dimensional one that retains meaningful characteristics of the original surface.

We again use a subspace discriminant ensemble to classify PIs by parameter. Table 2 compares these results to the same technique applied to low dimensional approximations of the raw surfaces and the normal distribution-based classifier built from surface variance alone. At each time in the system evolution, the best classification accuracy results from using PIs, improving accuracies over using either low resolution approximations of the surfaces or variance of surface height alone by at least 20%, including at early times in the evolution of the surface when pattern amplitudes are not visibly differentiated (see Figure 7). We postulate that PIs capture more subtle topological information that is useful for identifying the parameter used to generate each surface.

As we observed in §6.4.1, concatenating H_0 and H_1 PIs can notably improve the classification accuracy over either feature vector individually. We again note that classification accuracy appears insensitive to the PI parameters. For example, when the variance of the Guassians used to generate the PIs was varied from 0.0001 to 0.1, the classification accuracy of the H_0 PIs, changed by less than one percentage point. The classification accuracy for H_1 fluctuated in a range of approximately five points. For a fixed variance, when the resolution of the image was varied from 5 to 20, the H_0 accuracy varied by little more than three points until the accuracy dropped by six points for a resolution of 25.

PIs performed remarkably well in this classification task, allowing one to capitalize on subtle structural differences in the patterns and significantly reduce the dimension of the data for classification. There is more to be explored in the realm of pattern formation and persistence that is outside the scope of this paper.

7. Conclusion

PIs offer a stable representation of the topological characteristics captured by a PD. Through this vectorization, we open the door to a myriad of ML tools. This serves as a vital bridge between the fields of ML and topological data analysis and enables one to capitalize on topological structure (even in multiple homological dimensions) in the classification of data.

We have shown PIs yield improved classification accuracy over PLs and PDs on sampled data of common topological spaces at multiple noise levels using K-medoids. Additionally, computing distances between PIs requires significantly less computation time compared to computing distances between PDs, and comparable computation times with PLs. Through PIs, we have gained access to a wide variety of ML tools, such as SSVM which can be used for feature selection. Features (pixels) selected as discriminatory in a PI are interpretable because they correspond to regions of a PD. We have explored data sets derived from dynamical systems and illustrated that topological information of solutions can be used for inference of parameters since PIs encapsulate this information in a form amenable to ML tools, resulting in high accuracy rates for data that is difficult to classify.

The classification accuracy is robust to the choice of parameters for building PIs, providing evidence that it is not necessary to perform large-scale parameter searches to achieve reasonable classification accuracy. This indicates the utility of PIs even when there is not prior knowledge of the underlying data (i.e., high noise level, expected holes, etc.). The flexibility of PIs allows for customization tailored to a wide variety of real-world data sets.

Acknowledgments

We would like to acknowledge the research group of Paul Bendich at Duke University for allowing us access to a persistent homology package, which greatly reduced computational time and made analysis of large point clouds feasible. This code can be accessed via GitLab after submitting a request to Paul Bendich. This research is partially supported by the National Science Foundation under Grants No. DMS-1228308, DMS-1322508, NSF DMS-1115668, NSF DMS-1412674, NSF DMS-1615909, and DMR-1305449 as well as the DOD-USAF under Award Number FA9550-12-1-0408.

Appendix A. Homology and Data

Homology is an invariant that characterizes the topological properties of a topological space X. In particular, homology measures the number of connected components, loops, trapped volumes, and so on of a topological space, and can be used to distinguish distinct spaces from one another. More explicitly, the k-dimensional holes of a space generate a homology group, $H_k(X)$. The rank of this group is referred to as the *k-th Betti number*, β_k, and counts the number of k-dimensional holes of X. For a comprehensive study of homology, see the textbook of Hatcher (2002).

A.1 Simplicial Complexes and Homology

Simplicial complexes are one way to define topological spaces combinatorially. More precisely, a *simplicial complex* S consists of vertices (0-simplices), edges (1-simplices), triangles (2-simplices), tetrahedra (3-simplices), and higher-dimensional k-simplices (containing $k+1$ vertices), such that

- if σ is a simplex in S then S contains all lower-dimensional simplices of σ, and
- the non-empty intersection of any two simplices in S is a simplex in S.

The following setup is necessary for a rigorous definition of (simplicial) homology. To a simplicial complex, one can associate a chain complex of vector spaces over a field $\mathbb{F}$ (often a finite field $\mathbb{Z}/p\mathbb{Z}$ for p a small prime),

$$\cdots \to C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \to \cdots .$$

Here, vector space C_k consists of all $\mathbb{F}$-linear combinations of the k simplices of S, and has as a basis the set of all k-simplices. The linear map $\partial_k : C_k \to C_{k-1}$, known as the *boundary operator*, maps a k-simplex to its boundary, a sum of its $(k-1)$-faces. More formally, the boundary map acts on a k-simplex $[v_0, v_1, \ldots, v_k]$ by

$$\partial_k([v_0, v_1, \ldots, v_k]) = \sum_{i=0}^{k} (-1)^i [v_0, \ldots, \hat{v_i}, \ldots, v_k],$$

where $[v_0, \ldots, \hat{v_i}, \ldots, v_k]$ is the $(k-1)$-simplex obtained from $[v_0, \ldots, v_k]$ by removing vertex v_i. We define two subspaces of C_k: subspace $Z_k = \mathsf{ker}(\partial_k)$ is known as the *k-cycles*, and subspace $B_k = \mathsf{im}(\partial_{k+1}) = \partial_{k+1}(C_{k+1})$ is known as the *k-boundaries*. The boundary operator satisfies the property $\partial_k \circ \partial_{k+1} = 0$, which implies the inclusion $B_k \subseteq Z_k$.

Homology seeks to uncover an equivalence class of cycles that enclose a k-dimensional hole—that is, cycles which are not also boundaries of k-simplices. To this end, the k-th order homology is defined as $H_k(S) = Z_k/B_k$, a quotient of vector spaces. The k-th Betti number $\beta_k = \dim(H_k(S))$ is the dimension of this vector space, and counts the number of independent holes of dimension k. More explicitly, β_0 counts the number of connected components, β_1 the number of loops, β_2 the number of trapped volumes, and so on. Betti numbers are a topological invariant, meaning that topologically equivalent spaces have the same Betti numbers.

A.2 Persistence Diagrams from Point Cloud Data

One way to approximate the topological characteristics of a point cloud data set is to build a simplicial complex on top of it. Though there are a variety of methods to do so, we restrict attention to the *Vietoris–Rips simplicial complex* due to its computational tractability (Ghrist, 2008). Given a data set Y (equipped with a metric) and a scale parameter $\epsilon \geq 0$, the Vietoris–Rips complex S_ϵ has Y as its set of vertices and has a k-simplex for every collection of $k+1$ vertices whose pairwise distance is at most ϵ. However, it is often not apparent how to choose scale ϵ. Selecting ϵ too small results in a topological space with

a large number of connected components, and selecting ϵ too large results in a topological space that is contractible (equivalent to a single point).

The idea of persistent homology is to compute homology at many scales and observe which topological features persist across those scales (Ghrist, 2008; Carlsson, 2009; Edelsbrunner and Harer, 2008). Indeed, if $\epsilon_1 \leq \epsilon_2 \leq \ldots \leq \epsilon_m$ is an increasing sequence of scales, then the corresponding Vietoris–Rips simplicial complexes form a filtered sequence $S_{\epsilon_1} \subseteq S_{\epsilon_2} \subseteq \ldots \subseteq S_{\epsilon_m}$. As ϵ varies, so does the homology of S_ϵ, and for any homological dimension k we get a sequence of linear maps $H_k(S_{\epsilon_1}) \to H_k(S_{\epsilon_2}) \to \ldots \to H_k(S_{\epsilon_m})$. Persistent homology tracks the homological features over a range of values of ϵ. Those features which persist over a larger range are considered to be true topological characteristics, while short-lived features are often considered as noise.

For each choice of homological dimension k, the information measured by persistent homology can be presented as a *persistence diagram* (PD), a multiset of points in the plane. Each point $(x, y) = (\epsilon, \epsilon')$ corresponds to a topological feature that appears (is 'born') at scale parameter ϵ and which no longer remains ('dies') at scale ϵ'. Since all topological features die after they are born, this is an embedding into the upper half plane, above the diagonal line $y = x$. Points near the diagonal are considered to be noise while those further from the diagonal represent more robust topological features.

A.3 Persistence Diagrams from Functions

Let X be a topological space and let $f\colon X \to \mathbb{R}$ be a real-valued function. One way to understand the behavior of map f is to understand the topology of its sublevel sets $f^{-1}((-\infty, \epsilon])$, where $\epsilon \in \mathbb{R}$. Indeed, given $\epsilon_1 \leq \epsilon_2 \leq \ldots \leq \epsilon_m$, one can study map f using the persistent homology of the resulting filtration of topological spaces, known as the sublevel set filtration:

$$f^{-1}((-\infty, \epsilon_1]) \subseteq f^{-1}((-\infty, \epsilon_2]) \subseteq \ldots \subseteq f^{-1}((-\infty, \epsilon_m]).$$

If X is a simplicial complex, then one can produce an increasing sequence of simplicial complexes using a modification of this procedure called the lower star filtration (Edelsbrunner and Harer, 2010). Similarly, if X is a cubical complex (an analogue of a simplicial complex that is instead a union of vertices, edges, squares, cubes, and higher-dimensional cubes), then one can produce an increasing sequence of cubical complexes.

In §6.4.2, we study surfaces $u\colon [0,1]^2 \to \mathbb{R}$ produced from the Kuramoto-Sivashinsky equation. The domain $[0,1]^2$ is discretized into a grid of 512×512 vertices, i.e., a 2-dimensional cubical complex with 512^2 vertices, $511 \cdot 512$ horizontal edges, $511 \cdot 512$ vertical edges, and 511^2 squares. We produce an increasing sequence of cubical complexes as follows:

- A vertex v is included at scale ϵ if $u(v) \leq \epsilon$.
- An edge is included at scale ϵ if both of its vertices are present.
- A square is included at scale ϵ if all four of its vertices are present.

Our PDs are obtained by taking the persistent homology of the sublevel set filtration of this cubical complex.

We remark that PDs from point cloud data in §A.2 can be viewed as a specific case of PDs from functions. Indeed, given a data set X in some metric space (M, d), let $d_X \colon M \to \mathbb{R}$ be the distance function to set X, defined by $d_X(m) = \inf_{x \in X} d(x, m)$ for all $m \in M$. Note that $d_X^{-1}((-\infty, \epsilon])$ is the union of the metric balls of radius ϵ centered at each point in X. For $\epsilon_1 \leq \epsilon_2 \leq \ldots \leq \epsilon_m$, the persistent homology of

$$d_X^{-1}((-\infty, \epsilon_1]) \subseteq d_X^{-1}((-\infty, \epsilon_2]) \subseteq \ldots \subseteq d_X^{-1}((-\infty, \epsilon_m])$$

is identical to the persistent homology of a simplicial complex filtration called the *Čech complex*. Furthermore, the persistent homology of the Vietoris–Rips complex is an approximation of the persistent homology of the Čech complex (Edelsbrunner and Harer, 2010, Section III.2).

Appendix B. Examples of Persistence Images

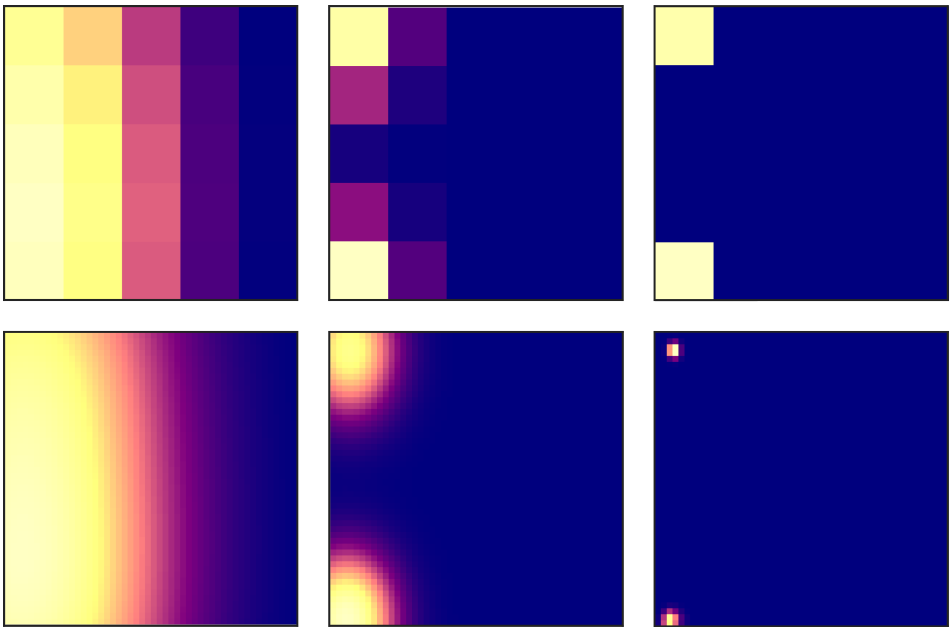

Figure 9: Examples of PIs for homology dimension H_1 arising from a noisy circle with a variety of resolutions and variances. The first row has resolution 5×5 while the second has 50×50. The columns have variance $\sigma = 0.2$, $\sigma = 0.01$, and $\sigma = 0.0001$, respectively.

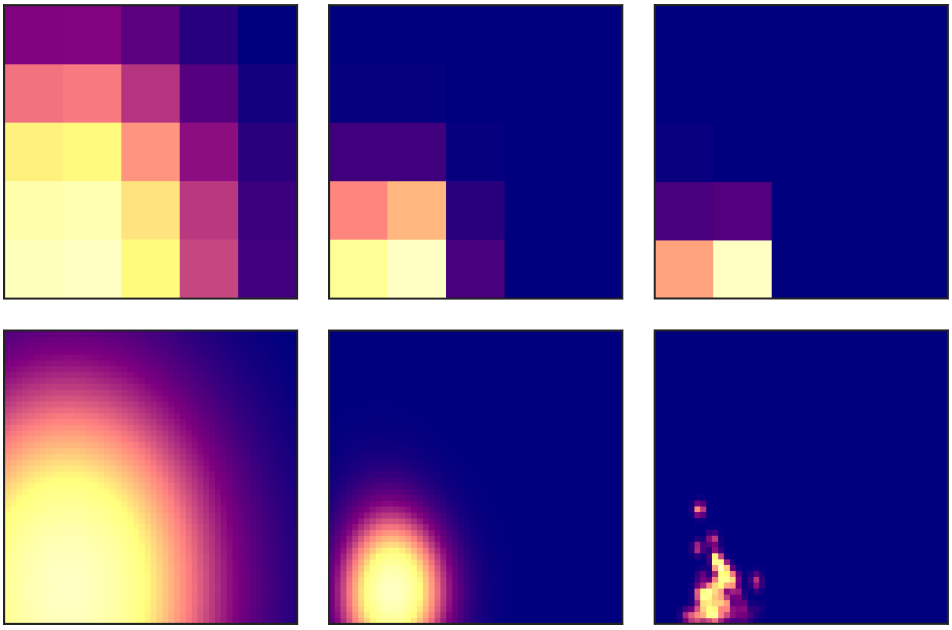

Figure 10: Examples of PIs for homology dimension H_1 arising from a noisy torus with a variety of resolutions and variances. The first row has resolution 5×5 while the second has 50×50. The columns have variance $\sigma = 0.2$, $\sigma = 0.01$, and $\sigma = 0.0001$, respectively.

Appendix C. Proofs of Equation (3) and Lemma 8

Let $u, v \in \mathbb{R}$ and $a, b > 0$. Equation (3) states that $\|ag_u - bg_v\|_1 = F(v-u)$, where $F\colon \mathbb{R} \to \mathbb{R}$ is defined by

$$F(z) = \begin{cases} |a-b| & \text{if } z = 0 \\ \left| a\text{Erf}\left(\frac{z^2+2\sigma^2\ln(a/b)}{z\sigma 2\sqrt{2}}\right) - b\text{Erf}\left(\frac{-z^2+2\sigma^2\ln(a/b)}{z\sigma 2\sqrt{2}}\right)\right| & \text{otherwise.} \end{cases}$$

Proof If $v = u$ then the statement follows from the fact that g_u and g_v are normalized to have unit area under the curve. Hence we may assume $u \neq v$.

For $u \neq v$ a straightforward calculation shows there is a unique real solution z^* to $ag_u(z) = bg_v(z)$, namely

$$z^* = \frac{v^2 - u^2 + 2\sigma^2 \ln(a/b)}{2(v-u)}.$$

Note

$$\|ag_u - bg_v\|_1 = \int_{-\infty}^{\infty} |ag_u(z) - bg_v(z)|dz = \left| \int_{-\infty}^{z^*} ag_u(z) - bg_v(z)dz + \int_{z^*}^{\infty} bg_v(z) - ag_u(z)dz \right|. \tag{5}$$

There are four integrals to compute, and we do each one in turn. We have

$$\begin{aligned}
\int_{-\infty}^{z^*} a g_u(z)\ dz &= \frac{a}{\sigma\sqrt{2\pi}} \int_{-\infty}^{z^*} e^{-(z-u)^2/2\sigma^2}\ dz \\
&= \frac{a}{\sqrt{\pi}} \int_{-\infty}^{P(v-u)} e^{-t^2}\ dt && \text{by substitution } t = \tfrac{z-u}{\sigma\sqrt{2}} \\
&= \frac{a}{\sqrt{\pi}} \left(\int_{-\infty}^{0} e^{-t^2}\ dt + \int_{0}^{P(v-u)} e^{-t^2}\ dt \right) \\
&= \frac{a}{\sqrt{\pi}} \left(\frac{\sqrt{\pi}}{2} + \frac{\sqrt{\pi}}{2} \mathrm{Erf}(P(v-u)) \right) \\
&= \frac{a}{2} \left(1 + \mathrm{Erf}(P(v-u))\right),
\end{aligned}$$

where $P(z) = \frac{z^2 + 2\sigma^2 \ln(a/b)}{z\sigma 2\sqrt{2}}$. Nearly identical calculations show

$$\begin{aligned}
\int_{z^*}^{\infty} a g_u(z)\ dz &= \frac{a}{2} \left(1 - \mathrm{Erf}(P(v-u))\right) \\
\int_{-\infty}^{z^*} b g_v(z)\ dz &= \frac{b}{2} \left(1 + \mathrm{Erf}(Q(v-u))\right) \\
\int_{z^*}^{\infty} b g_v(z)\ dz &= \frac{b}{2} \left(1 - \mathrm{Erf}(Q(v-u))\right),
\end{aligned}$$

where $Q(z) = \frac{-z^2 + 2\sigma^2 \ln(a/b)}{z\sigma 2\sqrt{2}}$. Plugging back into (5) gives $\|a g_u - b g_v\|_1 = F(v-u)$. ■

We now give the proof of Lemma 8.

Lemma 8. For $u, v \in \mathbb{R}^2$, let $g_u, g_v \colon \mathbb{R}^2 \to \mathbb{R}$ be normalized 2-dimensional Gaussians. Then

$$\|f(u) g_u - f(v) g_v\|_1 \leq \left(|\nabla f| + \sqrt{\frac{2}{\pi}} \frac{\min\{f(u), f(v)\}}{\sigma} \right) \|u - v\|_2.$$

Proof The result will follow from the observation that we can reduce the two-dimensional case involving Gaussians centered at $u, v \in \mathbb{R}^2$ to one-dimensional Gaussians centered at 0 and $r = \|u - v\|_2$. Let $u = (u_x, u_y)$ and $v = (v_x, v_y)$; we may assume $u_x > v_x$ w.l.o.g. Let (r, θ) be the magnitude and angle of vector $u - v$ when expressed in polar coordinates. The change of variables $(z, w) = R_\theta(x - v_x, y - v_y)$, where R_θ is the clockwise rotation of the

plane by θ, gives

$$
\begin{aligned}
&\|f(u)g_u - f(v)g_v\|_1 \\
=&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|\frac{f(u)}{2\pi\sigma^2}e^{-[(x-u_x)^2+(y-u_y)^2]/2\sigma^2} - \frac{f(v)}{2\pi\sigma^2}e^{-[(x-v_x)^2+(y-v_y)^2]/2\sigma^2}\right| dy\, dx \\
=&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left|\frac{f(u)}{2\pi\sigma^2}e^{-[w^2+(z-r)^2]/2\sigma^2} - \frac{f(v)}{2\pi\sigma^2}e^{-[w^2+z^2]/2\sigma^2}\right| dz\, dw \\
=&\int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}e^{-w^2/2\sigma^2}\left[\int_{-\infty}^{\infty}\left|\frac{f(u)}{\sigma\sqrt{2\pi}}e^{-(z-r)^2/2\sigma^2} - \frac{f(v)}{\sigma\sqrt{2\pi}}e^{-z^2/2\sigma^2}\right| dz\right] dw \\
=&\|f(u)g_r - f(v)g_0\|_1 \int_{-\infty}^{\infty}\frac{1}{\sigma\sqrt{2\pi}}e^{-w^2/2\sigma^2}dw \quad \text{with } g_0, g_r \text{ 1-dimensional Gaussians} \\
=&\|f(u)g_r - f(v)g_0\|_1 \\
\leq&|f(u)-f(v)| + \sqrt{\frac{2}{\pi}}\frac{\min\{f(u),f(v)\}}{\sigma}\|u-v\|_2 \quad \text{by Lemma 7} \\
\leq&\left(|\nabla f| + \sqrt{\frac{2}{\pi}}\frac{\min\{f(u),f(v)\}}{\sigma}\right)\|u-v\|_2.
\end{aligned}
$$

■

Appendix D. SSVM-based Feature Selection

We performed feature selection using one-against-all (OAA) SSVM on the six classes of synthetic data with noise level $\eta = 0.05$. The PIs used in the experiments were generated from the H_1 PDs, with the parameter choices of resolution 20×20 and variance $\sigma = 0.0001$. Note that because of the resolution parameter choice of 20×20, each PI is a vector in $\mathbb{R}^{400}$, and the selected features will be a subset of indices corresponding to pixels within the PI. We trained an OAA SSVM model for PIs of dimension H_1. In the experiment, we used 5-fold cross-validation and obtained 100% overall accuracy. Feature selection was performed by retaining the features with non-zero SSVM weights, determined by magnitude comparison using weight ratios (Chepushtanova et al., 2014). The resulting six sparse models contain subsets of discriminatory features for each class. Note that one can use only these selected features for classification without loss in accuracy. These features correspond to discriminatory pixels in the persistence images.

Figure 11 shows locations of pixels in the vectorized PIs selected by OAA SSVM that discriminate each class from all the others.

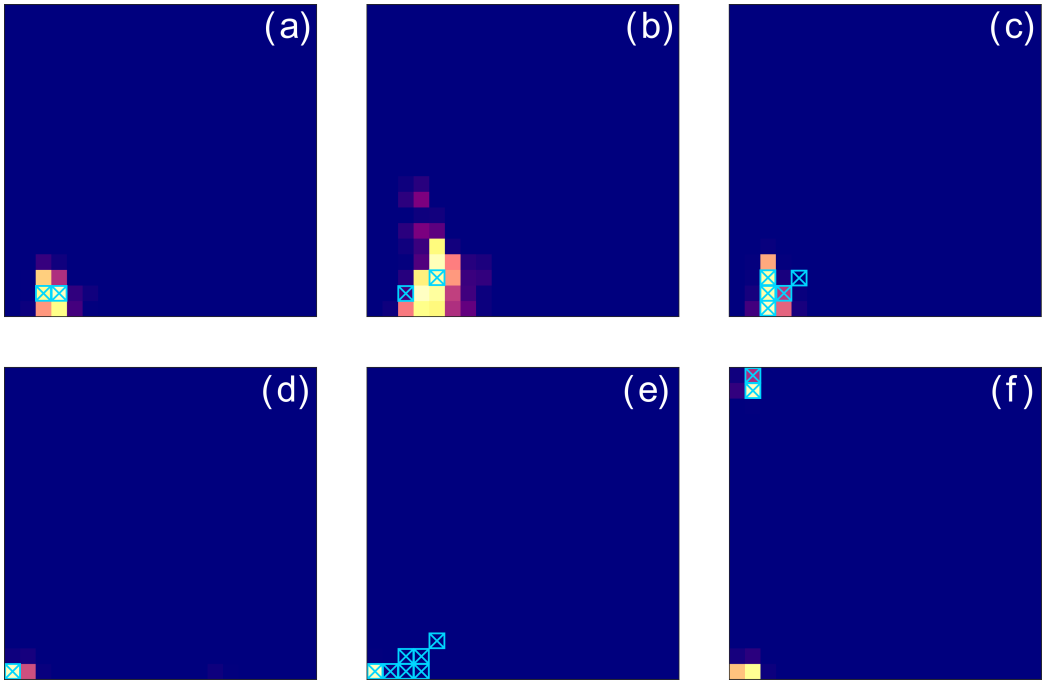

Figure 11: SSVM-based feature (pixel) selection for H_1 PIs from the six classes of the synthetic data. The parameters used are resolution 20×20 and variance 0.0001, for noise level 0.05. Selected pixels are marked by blue crosses. (a) A noisy solid cube with the two selected pixels (indices 59 and 79 out of 400). (b) A noisy torus with the two selected pixels (indices 59 and 98 out of 400). (c) A noisy sphere with the five selected pixels (indices 58, 59, 60, 79, and 98 out of 400). (d) Noisy three clusters with the one selected pixel (index 20 out of 400). (e) Noisy three clusters within three clusters with the seven selected pixels (indices 20, 40, 59, 60, 79, 80, and 98 out of 400). (f) A noisy circle with the two selected pixels (indices 21 and 22 out of 400).

References

Aaron Adcock, Erik Carlsson, and Gunnar Carlsson. The ring of algebraic functions on persistence bar codes. *Homology, Homotopy and Applications*, 18(1):381–402, 2016.

Paul Bendich. *Analyzing Stratified Spaces Using Persistent Versions of Intersection and Local Homology*. PhD thesis, Duke University, 2009.

Paul Bendich, James S. Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent homology analysis of brain artery trees. *Ann. Appl. Stat.*, 10(1):198–218, 2016.

Paul S. Bradley and Olvi L. Mangasarian. Feature selection via concave minimization and support vector machines. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 82–90, San Francisco, CA, 1998.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015.

Peter Bubenik and Pawel Dlotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 2016. Accepted.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46 (2):255–308, 2009.

Mathieu Carrière, Steve Y. Oudot, and Maks Ovsjanikov. Stable topological signatures for points on 3d shapes. In *Computer Graphics Forum*, volume 34, pages 1–12, 2015.

Frédéric Chazal, Vin de Silva, and Steve Oudot. Persistence stability for geometric complexes. *Geometriae Dedicata*, 173(1):193–214, 2014.

Yen-Chi Chen, Daren Wang, Alessandro Rinaldo, and Larry Wasserman. Statistical analysis of persistence intensity functions. *arXiv preprint arXiv:1510.02502*, 2015.

Sofya Chepushtanova, Christopher Gittins, and Michael Kirby. Band selection in hyperspectral imagery using sparse support vector machines. In *Proceedings SPIE DSS 2014*, volume 9088, pages 90881F–90881F15, 2014.

Moo K. Chung, Peter Bubenik, and Peter T. Kim. Persistence diagrams of cortical surface data. In *Information Processing in Medical Imaging*, pages 386–397. Springer, 2009.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.

David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. Lipschitz functions have L_p-stable persistence. *Foundations of computational mathematics*, 10(2): 127–139, 2010.

Rodolfo Cuerno and Albert-Lásló Barabási. Dynamic scaling of ion-sputtered surfaces. *Physical Review Letters*, 74:4746, 1995.

Yu Dabaghian, Facundo Memoli, Loren Frank, and Gunnar Carlsson. A topological paradigm for hippocampal spatial map formation using persistent homology. *PLoS Computational Biology*, 8(8):e1002581, 2012.

Barbara Di Fabio and Massimo Ferri. Comparing persistence diagrams through complex vectors. In *International Conference on Image Analysis and Processing 2015 Part I; Editors V. Murino, E. Puppo, LNCS 9279*, pages 294–305, 2015.

Pietro Donatini, Patrizio Frosini, and Alberto Lovato. Size functions for signature recognition. In *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation*, pages 178–183, 1998.

Herbert Edelsbrunner and John Harer. Persistent homology – a survey. *Contemporary Mathematics*, 453:257–282, 2008.

Herbert Edelsbrunner and John Harer. *Computational topology: An introduction*. American Mathematical Society, 2010.

Herbert Edelsbrunner, Alexandr Ivanov, and Roman Karasev. Current open problems in discrete and computational geometry. *Modelirovanie i Analiz Informats. Sistem*, 19(5): 5–17, 2012.

Tegan Emerson, Michael Kirby, Kelly Bethel, Anand Kolatkar, Madelyn Luttgen, Stephen O'Hara, Paul Newton, and Peter Kuhn. Fourier-ring descriptor to characterize rare circulating cells from images generated using immunofluorescence microscopy. *Computerized Medical Imaging and Graphics*, 40:70–87, 2015.

Massimo Ferri and Claudia Landi. Representing size functions by complex polynomials. *Proc. Math. Met. in Pattern Recognition*, 9:16–19, 1999.

Massimo Ferri, Patrizio Frosini, Alberto Lovato, and Chiara Zambelli. Point selection: A new comparison scheme for size functions (with an application to monogram recognition). In *Computer Vision ACCV'98*, pages 329–337. Springer, 1997.

Glenn M. Fung and Olvi L. Mangasarian. A feature selection newton method for support vector machine classification. *Computational Optimization and Applications*, 28(2):185–202, 2004.

Robert Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

Alexander A. Golovin and Stephen H. Davis. Effect of anisotropy on morphological instability in the freezing of a hypercooled melt. *Physica D: Nonlinear Phenomena*, 116:363–391, 1998.

Zhenhua Guo, Lei Zhang, and David Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.

Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

Kyle Heath, Natasha Gelfand, Maks Ovsjanikov, Mridul Aanjaneya, and Leonidas J Guibas. Image webs: Computing and exploiting connectivity in image collections. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Jan-Martin Hertzsch, Rob Sturman, and Stephen Wiggins. DNA microarrays: Design principles for maximizing ergodic, chaotic mixing. *Small*, 3(2):202–218, 2007.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

Leonard Kaufman and Peter Rousseeuw. *Clustering by means of medoids*. North-Holland, 1987.

Michael Kerber, Dmitriy Morozov, and Arnur Nigmetov. Geometry helps to compare persistence diagrams. In *2016 Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 103–112, 2016.

Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007, 2011.

Francis C. Motta, Patrick D. Shipman, and R. Mark Bradley. Highly ordered nanoscale surface ripples produced by ion bombardment of binary compounds. *Journal of Physics D: Applied Physics*, 45(12):122001, 2012.

Deepti Pachauri, Christian Hinrichs, Moo K. Chung, Sterling C. Johnson, and Vikas Singh. Topology-based kernels with application to inference problems in Alzheimer's disease. *IEEE Transactions on Medical Imaging*, 30(10):1760–1770, 2011.

Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, 2009.

Daniel A. Pearson, R. Mark Bradley, Francis C. Motta, and Patrick D. Shipman. Producing nanodot arrays with improved hexagonal order by patterning surfaces before ion sputtering. *Phys. Rev. E*, 92:062401, Dec 2015.

Jose A. Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, pages 1–40, 2013.

Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4741–4748, 2015.

Martin Rost and Joachim Krug. Anisotropic Kuramoto-Sivashinsky equation for surface growth and erosion. *Physical Review Letters*, 75:3894, 1995.

David Rouse, Adam Watkins, David Porter, John Harer, Paul Bendich, Nate Strawn, Elizabeth Munch, Jonathan DeSena, Jesse Clarke, Jeffrey Gilbert, Peter Chin, and Andrew Newman. Feature-aided multiple hypothesis tracking using topological and statistical behavior classifiers. In *SPIE Proceedings*, volume 9474, page 94740L, 2015.

Gurjeet Singh, Facundo Memoli, Tigran Ishkhanov, Guillermo Sapiro, Gunnar Carlsson, and Dario L. Ringach. Topological analysis of population activity in visual cortex. *Journal of Vision*, 8(8):11, 2008.

Chad M. Topaz, Lori Ziegelmeier, and Tom Halverson. Topological data analysis of biological aggregation models. *PloS One*, 10(5):e0126383, 2015.

Katharine Turner, Yuriy Mileyko, Sayan Mukherjee, and John Harer. Fréchet means for distributions of persistence diagrams. *Discrete & Computational Geometry*, 52(1):44–70, 2014.

Sara Kališnik Verovšek. Tropical coordinates on the space of persistence barcodes. *arXiv preprint arXiv:1604.00113*, 2016.

J. Villain. Continuum models of crystal growth from atomic beams with and without desorption. *J. Phys. I France*, 1:19–42, 1991.

Dietrich E. Wolf. Kinetic roughening of vicinal surfaces. *Physical Review Letters*, 67:1783, 1991.

Matthias Zeppelzauer, Bartosz Zieliński, Mateusz Juda, and Markus Seidl. Topological descriptors for 3d surface analysis. In *Computational Topology in Image Context: 6th International Workshop Proceedings*, pages 77–87, Marseille, France, 2016.

Li Zhang and Weida Zhou. On the sparseness of 1-norm support vector machines. *Neural Networks*, 23(3):373–385, 2010.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. *Advances in neural information processing systems*, 16(1):49–56, 2004.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.