Smooth Summaries of Persistence Diagrams and Texture Classification

Yu-Min Chung, and Michael Hull  
Department of Mathematics and Statistics  
University of North Carolina at Greensboro  
Greensboro, North Carolina 27412, USA  
{y_chung2, mbhull}@uncg.edu  

Austin Lawson  
Program of Informatics and Analytics  
University of North Carolina at Greensboro  
Greensboro, North Carolina 27412, USA  
azlawson@uncg.edu

Abstract

Topological data analysis (TDA) is a rising field in the intersection of mathematics, statistics, and computer science/data science. Persistent homology is one of the most commonly used tools in TDA, in part because it can be easily visualized in the form of a persistence diagram. However, performing machine learning algorithms directly on persistence diagrams is a challenging task, and so a number of summaries have been proposed which transform persistence diagrams into vectors or functions. Many of these summaries fall into the persistence curve framework developed by Chung and Lawson. We extend this framework and introduce new class of smooth persistence curves which we call Gaussian persistence curves. We investigate the statistical properties of Gaussian persistence curves and apply them to texture datasets: UIUCTex and KTH. Our classification results on these texture datasets perform competitively with the current state-of-arts methods in TDA.

1. Introduction

Topological Data Analysis (TDA) is a field of research lying at the intersection of mathematics, statistics, and computer science that is concerned with understanding data through its shape (see survey articles and references therein [8; 9; 21; 42; 13]). Driven by Algebraic Topology, this rapidly expanding subject has permeated several scientific disciplines, such as gene expression [39], aviation [30], and deep learning [25].

Persistent Homology, a tool in TDA, captures topological information by tracking changes in homological features over a filtration. It stores this information in a multi-set called a persistence diagram. Notably, there is a natural notion of distance called a $p$-Wasserstein distance between persistence diagrams with respect to which these diagrams are stable [17]; moreover, with the $p$-Wasserstein distance, the space of persistence diagrams is a metric space [31].

On the other hand, due to the multi-set structure, persistence diagrams are not easily compatible with many machine learning algorithms. Indeed, these algorithms are built on Hilbert spaces. Recent results have shown evidence that even when viewed as a metric space under the Wasserstein distances, the space of persistence diagrams fails to embed into a Hilbert space [7; 4]. Thus there is a need in the community to find useful summaries of persistence diagrams that are compatible with machine learning and also retain the topological information stored within them. The persistence landscape [6] is considered as one of the first attempts to transform persistence diagrams into scalar functions. Since then, there have been several other advancements in this direction, including persistent entropy [2; 3], persistence images [1], persistence indicator functions [37], template functions on persistence diagrams [41], persistence terrace [32], persistence B-spline grid vectors [19], persistence path [14], persistence codebooks [43], and several kernel based methods [35; 36; 27; 10; 29; 22]. Of particular interest to this paper is the persistence curve (PC) framework [16]. This is a general framework for creating functional summaries of persistence diagrams that encapsulates many of the examples mentioned above. In particular, persistence landscapes appear in the PC framework. The statistical properties of persistence landscapes are well studied [6; 12; 11]. This leads to the natural question: what conditions must one place on persistence curves in order to recover summaries with these useful statistical properties?

Our main contribution in this paper is to partially answer this question by proposing a new class of smooth summaries of persistence diagrams generated by the PC framework. The summary maps persistence diagrams to the space of smooth, integrable, real-valued functions by replacing points in a given diagram with Gaussian functions. This construction is similar to the construction of persistence surfaces in [1]; however, while they use this surface to define a collection of pixels that they call a persistence image, we instead integrate the surface over the quadrant whose lower right corner intersects the diagonal at $(t, t)$ to produce a summary which is a smooth function of $t$. We refer to the
summaries constructed in this way as Gaussian persistence curves. These summaries naturally live in a Hilbert space of absolutely integrable functions and hence can be used as inputs for a variety of machine learning techniques. To the best of authors’ knowledge, the proposed summaries are some of the first smooth functional summaries.

These smooth summaries have a number of both theoretical and practical advantages, and exploring all of these is part of a larger work in progress. Here, we focus on statistical properties of smooth persistence curves. Our main theoretical result is a form of the central limit theorem for Gaussian persistence curves (Theorem 1). Similar statistical results for persistence landscapes and other functional summaries appear in [12; 5]. We then use synthetic data to illustrate the fact that our curves can distinguish different spaces using only points sampled with heavy noise from those spaces. Finally, we test our Gaussian persistence curves on the problem of classifying grey-scale images according to texture. We use two popular texture databases, UIUC-Tex [33] and KTH-TIPS2b [23]. We find that the Gaussian persistence curves are competitive with the persistence summaries constructed in this way as Gaussian persistence curves. These summaries naturally live in a Hilbert space of absolutely integrable functions and hence can be used as inputs for a variety of machine learning techniques. To the best of authors’ knowledge, the proposed summaries are some of the first smooth functional summaries.

We structure the paper as follows. In Section 2, we give a light introduction to homology by way of cubical sets and persistent homology while referring the reader to [24] and [20] for more information on these two subjects respectively. Note that this is purely for instructive reasons. As we will see, the PC framework is defined on the space of persistence diagrams and makes no assumptions about the underlying homology theory.

A cubical set is a set $X$ that can be written as a finite union of cubes whose vertices lie on an integer lattice. For example, an image is a type of cubical set that can be described entirely by its two dimensional cells. Given a ring $R$ (often taken to be $\mathbb{Z}_2$), the $k$-th chain space $C_k(X; R)$ is a free abelian group generated by the $k$-cubes of $X$ with coefficients in $R$. For each $k$ there is a natural map called a boundary map $\partial_k$ that sends elements in $C_k(X; R)$ to $C_{k-1}(X; R)$ is such a way that $\partial_{k-1} \partial_k \equiv 0$. This property of the boundary map allows us to define the $k$-th homology group, which is the quotient $H_k(X; R) = \ker \partial_k / \operatorname{im} \partial_{k+1}$.

The $k$-th Betti number $\beta_k(X)$ is defined to be the rank of the $k$-th homology group. We remind the reader that cubical homology is one of many homology theories one can use to compute homology, and proceeding from here we only assume that the homology of a given space is defined.

A filtration of a topological space $X$ is a sequence of subspaces of $X$, $\emptyset = X_0 \subset X_1 \subset \cdots \subset X_n = X$. Applying homology to this sequence leads to a sequence of groups $H_k(X_i)$ with homomorphisms induced by inclusion $f_k^{i,i+1} : H_k(X_i) \to H_k(X_{i+1})$. We define the map $f_k^{i,j} : H_k(X_i) \to H_k(X_j)$ by composition of subsequent maps when $j > i$. The ranks of the groups $\operatorname{rank} f_k^{i,j}$ with $d \geq b$ form the persistent Betti numbers $\beta_k^{b,d}$. We say a homology class $\alpha$ is born at $b$ if $\alpha \in H_k(X_b)$ and $\alpha \notin \operatorname{im} f_k^{b+1,b}$.

We alpha dies at $d$ if $\alpha \notin \operatorname{im} f_k^{d,b}$ and $\alpha \in \operatorname{im} f_k^{d-1,b}$. We can count the multiplicity of a birth death pair by using the inclusion-exclusion principle: $\xi_k^{b,d} = \beta_k^{d-1,b} - \beta_k^{b+1,d-1} + \beta_k^{b+1,d} - \beta_k^{b,d}$. We can store this birth-death information along with multiplicities $\xi_k^{b,d}$ in a multi-set called the $k$-th dimensional persistence diagram in which we also include infinitely many copies of the diagonal $\{(x,x) \in \mathbb{R}^2\}$. Finally, the Fundamental Lemma of Persistent Homology (FLPH) [20] states that for a persistence diagram $D$ arising from a filtration, the $k$-th Betti number of the $t$-th member can be obtained by counting the number of diagram points (birth-death pairs) that lie within the upper left quadrant whose lower right corner lies at $(t,t)$, or more precisely, $\beta_k(X_t)$ is given by the sum $\beta_k(X_t) = \sum_{b \leq t < d, (b,d) \in D} \xi_k^{b,d}$.

2.2. Images

Our main application to this paper is texture classification in images. Let $[n] = \{0, 1, \ldots, n - 1\}$. Formally, an $m \times n$ binary image is a function $I : [m] \times [n] \to \{0, 1\}$.

A pair $(i, j)$ in the domain of $I$ is called a pixel and $I(i,j)$ is called a pixel value. In this paper, we associate the color white for a binary pixel value of 1 and black for a value of 0. We can treat a binary image as a cubical set by considering the collection of its white pixels. In this way, we can compute components ($H_0$) by counting the number of clusters of white pixels (using the notion of 4-connectivity in images) and we can compute holes ($H_1$) by counting the clusters of black pixels that are surrounded completely by
white pixels. Figure 1 displays two binary images that illustrate this. In (a) we can see four components ($\beta_0 = 4$), and one hole ($\beta_1 = 1$). However, by visual inspection is seems as if there may be another hole. This phenomenon is called the boundary effect. To account for this, we also consider the inverse of the binary image, that is the image with the pixel values flipped, as shown in Figure 1(b). This image has three components (including the background) and three holes. Taking the information from an image and its inverse gives us a clearer picture of the true homological nature of the depicted object.

We are interested in classifying textures from grayscale images. An $m \times n$ grayscale image is a function

$$I : [m] \times [n] \to [256]$$

The inverse of a grayscale image $I$ is the image $I^C = 255 - I$. We cannot easily compute homology on a grayscale image as this will require assigning it a cubical set. This can be done by thresholding the image at some value $t \in [256]$ to produce the binary image $I_t(i,j)$ that is $1$ if $I(i,j) \leq t$ and $0$ otherwise. However, this requires a choice of $t$. Instead, we will assign the sequence of all possible binary images obtained by thresholding: $I_0 \leq I_1 \leq \ldots \leq I_{255}$. This generates a filtration of the corresponding cubical sets, which then allows us to compute persistent homology and obtain a persistence diagram. Because images are two dimensional, we are only interested in 0 and 1-dimensional persistence diagrams. We end this section with a small example of the persistent homology process.

Example 1. Consider the following grayscale image:

$$I = \begin{bmatrix} 1 & 3 & 2 \\ 1 & 10 & 2 \\ 1 & 3 & 2 \end{bmatrix}$$

It is easy to see the threshold values of interest here are 1, 2, 3, and 10. We consider these threshold values in sequence and track the changes in homology. Recall that thresholding creates a binary where white represents a pixel of $I$ with a value below the threshold and black represents otherwise.

- $t = 1$: $I_1 = \begin{bmatrix} \hline \hline \hline \end{bmatrix} \Rightarrow \beta_0 = 1, \beta_1 = 0$. A $\beta_0$ generator is born at 1.
- $t = 2$: $I_2 = \begin{bmatrix} \hline \hline \hline \end{bmatrix} \Rightarrow \beta_0 = 2, \beta_1 = 0$. A $\beta_0$ generator is born at 2.
- $t = 3$: $I_3 = \begin{bmatrix} \hline \hline \hline \end{bmatrix} \Rightarrow \beta_0 = 1, \beta_1 = 1$. A $\beta_0$ generator dies at 3 and a $\beta_1$ generator is born at 3.
- $t = 10$: $I_{10} = \begin{bmatrix} \hline \hline \hline \end{bmatrix} \Rightarrow \beta_0 = 1, \beta_1 = 0$. The $\beta_1$ generator dies at 10 and the $\beta_0$ generator persists.

The elder rule tells us that the $\beta_0$ generator that died at $t = 3$ in this case is the younger one, i.e. the one born at 2. Collecting this information, we extract the 0 and 1-dimensional persistence diagrams, identified by the non-diagonal points, $D_0 = \{(1, \infty), (2,3)\}$ and $D_1 = \{(3,10)\}$.

3. The Persistence Curve Framework

The FLPH indirectly states that the Betti number of the $t$-th member of the filtration is found by counting the number of off diagonal points in a diagram with multiplicity that lie inside the fundamental box at $t$, $F_t = \{(x,y) \mid x \leq t < y\}$. The persistence curve framework describes the fundamental box described in the FLPH to generate functions from persistence diagrams. Let $D$ be the set of all persistence diagrams, $\Psi$ be the set of all functions $\psi : D \times \mathbb{R}^2 \to \mathbb{R}$ with $\psi(D;x,x,t) = 0$ for all $(x,x) \in \mathbb{R}^2$ and $D \in D$. Let $\mathcal{R}$ represent the set of functions on $\mathbb{R}$. To ease the notation, we will often refer to $\psi(D;x,y,t)$ as $\psi(x,y,t)$ when $D$ is understood. Moreover, when $\psi$ does not depend on $t$, we denote it by $\psi(x,y)$. Let $\mathcal{T}$ be a set of operators $T(S,f)$ that read in a multi-set $S$ and real-valued function $f$ and returns a scalar. For example, $T(S,f) = \max_s \{ f(s) \mid s \in S \}$ is the k-max operator, i.e. the operator that returns the k-largest element of a set.

Definition 1. We define a map $P : D \times \Psi \times \mathcal{T} \to \mathcal{R}$ where

$$P(D,\psi,T)(t) := T(F_t, \psi(D;x,y,t)), \quad t \in \mathbb{R}$$

The function $P(D,\psi,T)$ is called a persistence curve on $D$ with respect to $\psi$ and $T$.

Definition 1 is a more general version than the one proposed in [16], which, for some function $Q \in \mathcal{T}$ that maps multi-sets to real numbers, defined $T(S,f) = Q \circ f(D \cap S)$. The definition proposed here drops the requirement to apply a function on only the diagram points lying in $S$ thus allowing for general integration. For example, let $\eta$ be
a measure on $\mathbb{R}^2$. If $\psi(D; x, y, t) = \psi(D; x, y)$ is integrable with respect to $\eta$ for each diagram $D$. We can define $T(F_t, \psi) = \int_{F_t} \psi \, d\eta$. With this form, the sum statistic that appears in [16], $\sum(\psi(D; F_t))$ can be rewritten as $\int_{F_t} \psi \, d\#$ where $\#$ is the counting measure. We provide a couple examples below.

Example 2. Given a diagram $D$, The Betti curve $\beta_D$ is the curve generated by FLPH. In the framework of persistence curves it uses the sum statistic $T(S, f) = \int_S f \, d\# := \int d\#$ and the function $\psi(x, y) = \chi_D(x, y)$ the indicator function on the points of the diagram

$$\beta_D(t) = P \left( D, \chi_D, \int d\# \right).$$

Example 3. Given a diagram $D$, we can define the life curve $\ell_D$ by taking $\psi(D; x, y) = \ell(D; x, y) := (y - x) \cdot \chi_D(x, y)$ We use the sum statistic $T(S, f) = \int_S f \, d\# := \int d\#$ and define

$$\ell_D(t) = P \left( D, \ell, \int d\# \right).$$

Example 4. Given a persistence diagram $D$, define for $(b, d) \in D$,

$$l(b, d, t) = \begin{cases} 0 & \text{if } t \notin (b, d) \\ t - b & \text{if } t \in (b, \frac{b+d}{2}] \\ d - t & \text{if } t \in (\frac{b+d}{2}, d] \\ \end{cases}.$$

If $(b, d) \notin D$ we define $l(b, d, t) = 0$. Then the $k$-th Persistence Landscape [6] is defined by $\lambda_k(t) = \max_k \{ l(b, d)(t) | (b, d) \in D \}$. By taking $T(S, f) = \max_k \{ f(s) | s \in S \}$, recover the $k$-th landscape as the persistence curve $P(D, l, t) \equiv \lambda_k$.

4. Smooth Persistence Curves

Definition 2. Fix $\psi$ and $T$. If the derivative of $P(D, \psi, T)(t)$ exists and is continuous, i.e. $P(D, \psi, T)(t) \in C^1(\mathbb{R})$, for every diagram $D \in \mathcal{D}$, then we call $P(\cdot, \psi, T)$ a smooth persistence curve.

Next, we describe a general procedure for generating smooth persistence curves by centering a Gaussian function at every point. This will allow us to create smooth versions of the curves found in [16].

Let $D$ be a diagram and $\Sigma$ be a symmetric, positive semidefinite $2 \times 2$ matrix. For a point $\mu \in \mathbb{R}^2$, let $g_{\mu, \Sigma}$ be the probability density function (PDF) of a bivariate normal distribution with mean $\mu$ and covariance matrix $\Sigma$. That is, $g_{\mu, \Sigma}(x) = \frac{\exp \left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) \right)}{2\pi |\Sigma|^{1/2}}$. Finally, let $m$ be the Lebesgue measure on $\mathbb{R}^2$.

Definition 3. Let $\kappa(D; b, d)$ be a real valued function with $\kappa(D; b, b, t) = 0$. A Gaussian persistence curve is a persistence curve of the form

$$P \left( D, \sum_{(b, d) \in D} \kappa(D; b, d)g_{(b, d), \Sigma}, \int dm \right).$$

As it turns out, this definition not only leads to a smooth persistence curve, but a well-controlled summary with respect to to input $t \in \mathbb{R}$.

Proposition 1. Suppose $\kappa(D; b, d)$ is a real-valued function so that $\kappa(D; b, b) = 0$. Moreover, suppose $P \left( D, \sum_{(b, d) \in D} \kappa(D; b, d)g_{(b, d), \Sigma}, \int dm \right)$ is a Gaussian persistence curve. Then $P$ is $k$-Lipschitz with $k \leq \sum_{(b, d) \in D} |\kappa(b, d)|$.

Proof. Let $g(x, y)$ be the PDF of a bivariate normal distribution. We will prove the derivative of $P$ is uniformly bounded. By applying Leibniz’s integral rule and Fundamental Theorem of Calculus, we achieve

$$\frac{d}{dt} \int_t^\infty \int_{-\infty}^t g(x, y) \, dx \, dy = \int_t^\infty g(t, y) \, dy + \int_t^\infty \frac{\partial}{\partial t} \int_t^\infty g(x, y) \, dy \, dx = \int_t^\infty g(t, y) \, dy + \int_t^\infty \frac{\partial}{\partial t} \int_{-\infty}^t g(x, t) \, dx \, dy.$$

Then we see

$$\frac{d}{dt} P(t) = \frac{d}{dt} \int_t^\infty \int_{-\infty}^t \sum_{(b, d) \in D} \kappa(b, d)g_{(b, d), \Sigma}(x, y) \, dx \, dy$$

$$\leq \sum_{(b, d) \in D} \kappa(b, d) \left( \int_t^\infty g(t, y) \, dy - \int_{-\infty}^t g(x, t) \, dx \right).$$

We will give a few examples of Gaussian PCs below.

Example 5. Let $\kappa(D; b, d) = 1$. Let $I_2$ be the $2 \times 2$ identity matrix and let $\sigma^2 > 0 \in \mathbb{R}$. Then with $\Sigma = \sigma^2 \cdot I_2$. The resulting Gaussian persistence curve $P(t)$ is given by

$$P(t) = \int_t^\infty \int_{-\infty}^t \sum_{(b, d) \in D} g_{(b, d), \Sigma}(x, y) \, dx \, dy.$$
Because $\Sigma$ is diagonal, we can split $g_{b,d,\Sigma} = g_{b,\sigma^2}(x)g_{d,\sigma^2}(y)$. This means we can rewrite $P$ as

$$P(t) = \sum_{(b,d) \in D} G_{b,\sigma^2}(t)(1 - G_{d,\sigma^2}(t)).$$

Example 5 can be viewed as a smooth version of the Betti curve. The Euler Characteristic of a complex is defined to be the alternating sum of its Betti numbers. The Euler Characteristic Curve (ECC) of a filtration is the sequence of Euler Characteristics of the complexes in the filtration. In the PC framework, the ECC is defined as the alternating sum of the Betti curves. Thus we can define a smooth ECC as an alternating sum of smooth Betti curves. A smooth Euler Characteristic Curve is defined in [18] by calculating the mean of the original ECC, subtracting that value from the original ECC and then integrating the resulting function with respect to time (the filtration sequence). Though we can define the smooth ECC, it is typically better to its summands separately. Using the ECC itself can lead to a decrease in performance.

We can obtain smooth versions of other curves appearing in [16].

**Example 6.** Let $\ell$ be defined as in Example 3 and let $\ell_{sum} = \sum_{(b,d) \in D} \ell(b, d)$. Let $n$ be given and let $\Sigma = \sigma^2 \cdot I$.

Define $\kappa(D, x, y, t) = \frac{\ell_{sum}}{\ell_{sum}}$. Then we can define a smooth version of the life curve by $P(D, \psi, \int dm)$. Because the points in the diagram are independent of $P(D, \psi, \int dm)$, we can see this function has a bounded derivative and hence is Lipschitz.

We can use the idea of Example 6 to generate similar curves for other functions such as the midlife function $\frac{b+d}{2}$, entropy function $\log \log \frac{b+d}{d-b}$, and multiplicative life function $\frac{d-b}{d}$ among others. We also note that we take $\Sigma$ as a scalar multiple of the identity often in practice. For applications in this paper, we will use two curves which we call the Gaussian life curve (gl$_{\sigma}$) and the Gaussian midlife curve (gml$_{\sigma}$) for $\sigma > 0$:

$$g_{l\sigma} = P(D, \sum_{(b,d) \in D} \frac{\ell(b, d)}{\ell_{sum}} g_{b,d,\sigma^2,I_2}, \int dm)$$

(1)

$$g_{m\sigma} = P(D, \sum_{(b,d) \in D} \frac{b+d}{m_{sum}} g_{b,d,\sigma^2,I_2}, \int dm)$$

(2)

where $m_{sum} = \sum_{(b,d) \in D} (b + d)$.

5. **Stochastic convergence of persistence curves**

While persistence curves can defined on the space $D$ of all possible persistence diagrams, in this section we need to restrict to the space $D_N$ of persistence diagrams $D$ with at most $N$ points and with the property that $|x| \leq N$ and $|y| \leq N$ for all $(x, y) \in D$. We consider $\psi$ and $T$ to be fixed and the corresponding persistence curve $P$ to be a map from $D_N$ to $R$. We will assume that $\psi$ and $T$ are such that

$$\sup_{D \in D_N, t \in I} \{P(D, \psi, T)(t)\} < \infty$$

where $I = [-N, N]$. For all of the $\psi$ and $T$ we consider this condition will be satisfied.

Let $P$ be a probability distribution on $D_N$. The expectation $\mu$ of the random variable $P$ with respect to $P$ is called the average persistence curve. Now let $D_1, ..., D_n$ be a sample with respect to the distribution $P$. define $P_i = P(D_i, \psi, T)$. The empirical average persistence curve is $P_n(t) := \frac{1}{n} \sum_{i=1}^n P_i(t)$.

For a fixed $t \in I$, it follows from the law of large numbers that $P_n(t)$ converges to $\mu(t)$ almost surely and from the central limit theorem that $\sqrt{n}(P_n(t) - \mu(t))$ converges in distribution to a mean zero normal random variable with the same variance as $P_n$. We will show that this convergence is in fact uniform with respect to the variable $t$.

Let $f_i : D_N \rightarrow R$ be defined by $f_i(D) = P(D, \psi, T)(t)$, and let $F = \{f_i \mid i \in I\}$. We will show that $\sqrt{n}(P_n(t) - \mu(t))$ converges weakly to a Gaussian process on $F$. Here a Gaussian process is a stochastic process indexed by $t \in I$ such that for any finite set of indices $t_1, ..., t_n$, $(f_{t_1}, ..., f_{t_n})$ is a multivariate Gaussian random variable on $D_N$. $X_n$ converges weakly to $X$ means that for every bounded continuous $f, E^*(f(X_n)) \rightarrow E(f(X))$, where $E^*$ denotes outer expectation, which is similar to expectation but allows for the possibility that $f(X_n)$ may not be measurable.

$F$ is called an envelope for $F$ if $|f_i(D)| \leq F(D)$ for all $f_i \in F$ and all $D \in D_N$. If $Q$ is a probability measure on $D_N$, then $N(\varepsilon, F, L_r(Q))$ is the minimum number of $\varepsilon$-balls needed to cover $F$ with respect to the norm $\|f\|_{Q,r} := \left( \int f^* dQ \right)^{\frac{1}{r}}$. Define

$$J(\delta, F, L_r) := \int_0^\delta \sqrt{\log \sup_Q N(\varepsilon||F||_{Q,r}, F, L_r(Q))} d\varepsilon,$$

where the supremum is taken over all finitely discrete probability measures $Q$ on $D_N$ and $F$ is an envelope for $F$.

Our proof of convergence is similar to the proof for persistence landscapes which appears in [12]. In particular, the proof is based on [26, Theorem 2.5]. A similar result also appears in a more general context in [5].

**Theorem 1.** Let $\psi$ and $T$ be fixed, and suppose that there exists $k$ such that $P(D, \psi, T)$ is $k$-Lipschitz for all $D \in D_N$. Then

$$\sqrt{n}(P_n(t) - \mu(t))$$
weakly converges to a mean zero Gaussian process on $\mathcal{F}$ with covariance $\int fg d\mathbb{P} - \int f d\mathbb{P} \int g d\mathbb{P}$.

Proof. Define $F(D) = \sup_{t \in \mathbb{T}} f_t(D)$, which is an envelope for $\mathcal{F}$. By assumption $\sup_{D \in \mathcal{D}_N} F(D) < \infty$, and hence $\int F^2 d\mathbb{P} < \infty$. In order to apply [26, Theorem 2.5], it only remains to show that $J(1, \mathcal{F}, L_2) < \infty$.

Fix $0 < \varepsilon < 1$ and a finitely discrete probability measure $Q$ on $\mathcal{D}_N$. Choose $-N = t_0 < t_1 < \ldots < t_m < t_{m+1} = N$ such that $|t_i - t_{i-1}| \leq \frac{\varepsilon}{2} ||F||_{Q,2}$ for all $1 \leq i \leq m + 1$ and $m = \frac{2kN}{\varepsilon||F||_{Q,2}}$. We claim that the set of $\varepsilon||F||_{Q,2}$-balls centered at $f_{t_0}, f_{t_1}, \ldots, f_{t_m}$ covers $\mathcal{F}$. Let $f_t \in \mathcal{F}$, and suppose $t_{i-1} \leq t \leq t_i$. Since each persistence curve $P$ is $k$-Lipschitz, $||f_{t_{i-1}} - f_t||_{Q,2} \leq k|t_{i-1} - t| \leq \varepsilon||F||_{Q,2}$ and similarly for $||f_{t_i} - f_t||_{Q,2}$. It follows that $\sup_Q N(\varepsilon||F||_{Q,2}, \mathcal{F}, L_2(Q)) \leq \frac{2kN}{\varepsilon||F||_{Q,2}}$, and hence $J(1, \mathcal{F}, L_2) = \int_0^1 \log \sup_Q N(\varepsilon||F||_{Q,2}, \mathcal{F}, L_2(Q)) d\varepsilon \leq \int_0^1 \log(\frac{2kN}{\varepsilon||F||_{Q,2}}) d\varepsilon < \infty$. The theorem now follows from [26, Theorem 2.5].

To conclude this section, we explored the convergence properties of the Gaussian life curve defined in Example 6 via synthetic data. We also examined its capability to distinguish spaces with both synthetic and real data. With regards to synthetic data we consider the three spaces shown in Figure 2. We used scikit-tda’s TaDaSets [38] package to draw the synthetic samples presented here. Each sample of a circle contains 50 points drawn from a unit circle with Uniform[-0.15,0.15] noise. Similarly, the figure 8 space is sampled with the same noise. Finally, the Swiss Roll space is sampled with 0.8 noise and then coordinates are divided by 10 to match the scale of the circle and figure 8 spaces. Figure 3 demonstrates the convergence of the Gaussian life persistence curve with covariance matrix $I_2$. Each plot shows twenty averages taken on $n$ samples where $n \in \{10, 50, 100, 200\}$. The samples were drawn via TaDaSets’s dSphere function and diagrams calculated with Ripser [40]. The curves were calculated on the 1-dimensional diagram for each sample via the PersistenceCurves [28] package. Figure 4 shows a plot of the average of Gaussian life persistence curves 20 each for the unit circle space, figure 8 space, and Swiss Roll space. For each space, we used a covariance of $0.1 \cdot I_2$.

6. Application to Texture Classification

We tested the performance of Gaussian persistence curves on two popular texture databases, UIUCTex [33], which contains 1000 480 by 640 grayscale images in 25 different classes, and KTH Textures under varying Illumination, Pose and Scale (KTH-TIPS2b) containing 810 200 by 200 grayscale images in 10 different classes [23]. For each of the databases we mimicked the score calculation found in [35] which produced a 100 random 80/20 stratified train-test splits and averaged the classification accuracies. Our models consisted of the concatenation of four vectors generated by Gaussian life persistence curve ($gl_1$) computed on the 0 and 1-dimensional diagrams of an image and its inverse over the values $t \in \{0, 1, \ldots, 255\}$. This results in four 256-dimensional vectors, and the concatenation...
displays the results of these tests along with that utilized large margin nearest neighbors. The set of statistics is mean, standard deviation, and the normalized life ($sl$) and normalized midlife ($sml$) curves that appear in persistence diagrams by utilizing the persistence curve concatenation of curves based on the midlife function ($sml$). For each diagram $D$, we integrate these Gaussians over the fundamental box at $t$. This process maps persistence diagrams to smooth, absolutely integrable, Lipschitz functions. We proved that the sample mean distribution of Lipschitz continuous persistence curves (hence the Gaussian PCs) weakly converges to a Gaussian process. These curves proved successful and competitive with other TDA methods in the task of texture classification. The Gaussian PCs are one example of many summaries one can derive from the PC framework. The richness of PCs opens a door to several future directions of expansion for the theory around the framework such as bootstrapping, hypothesis testing, and stability analysis.

## References


<table>
<thead>
<tr>
<th>Model</th>
<th>UIUCTex</th>
<th>KTH-TIPS2b</th>
</tr>
</thead>
<tbody>
<tr>
<td>gl+RF</td>
<td>93.2 ± 1.8%</td>
<td>92.5 ± 2.0%</td>
</tr>
<tr>
<td>gl+PS+RF</td>
<td>94.1 ± 1.6%</td>
<td>94.7 ± 1.9%</td>
</tr>
<tr>
<td>gml+RF</td>
<td>91.7 ± 1.8%</td>
<td>91.4 ± 2.3%</td>
</tr>
<tr>
<td>gml+PS+RF</td>
<td>92.8 ± 1.8%</td>
<td>94.1 ± 1.7%</td>
</tr>
<tr>
<td>gl+gml+RF</td>
<td>93.1 ± 1.7%</td>
<td>92.3 ± 2.1%</td>
</tr>
<tr>
<td>gl+gml+PS+RF</td>
<td>93.8 ± 1.6%</td>
<td>94.2 ± 1.8%</td>
</tr>
<tr>
<td>gECC + RF</td>
<td>82.0 ± 2.2%</td>
<td>90.3 ± 2.2%</td>
</tr>
<tr>
<td>gECC + PS + RF</td>
<td>90.3 ± 2.0%</td>
<td>95.7 ± 1.6%</td>
</tr>
<tr>
<td>sl + RF [16]</td>
<td>93.1 ± 1.8%</td>
<td>94.2 ± 1.7%</td>
</tr>
<tr>
<td>sl + PS + RF [16]</td>
<td>94.1 ± 1.6%</td>
<td>96.1 ± 1.8%</td>
</tr>
<tr>
<td>sml + RF [16]</td>
<td>91.2 ± 1.8%</td>
<td>91.3 ± 2.4%</td>
</tr>
<tr>
<td>sml + PS + RF [16]</td>
<td>92.4 ± 2.0%</td>
<td>94.4 ± 1.9%</td>
</tr>
<tr>
<td>ECC + RF [16]</td>
<td>81.4 ± 2.3%</td>
<td>89.4 ± 2.7%</td>
</tr>
<tr>
<td>ECC + PS + RF [16]</td>
<td>91.0 ± 1.9%</td>
<td>90.1 ± 2.7%</td>
</tr>
<tr>
<td>EKFC-LMNN [35]</td>
<td>91.23 ± 1.1%</td>
<td>94.77 ± 1.3%</td>
</tr>
<tr>
<td>PI[1] + RF</td>
<td>91.5 ± 2.0%</td>
<td>86.3 ± 2.5%</td>
</tr>
</tbody>
</table>

Table 1. Performances of various Gaussian persistence curves.

Finally, we fed each of the models to scikit-learn's random forest (RF) algorithm for training and classification. Table 1 displays the results of these tests along with the results of the EKFC+LMNN a klein bottle-based model that utilized large margin nearest neighbors [35]. We also calculated the scores of the normalized life ($sl$) and normalized midlife ($sml$) curves that appear in [16]. For the persistence image (PI) calculations, the PIs were calculated on the same four diagrams previously mentioned. The resulting PIs were flattened into vectors, concatenated, then fed into the random forest algorithm. In this table, we see competitive scores among the curves, particularly between the gl+PS+RF, sl+PS+RF, and EKFC+LMNN models.

## 7. Conclusion

This paper proposed a new class of summary functions for persistence diagrams by utilizing the persistence curve framework. In essence, this class replaces the points of a diagram with weighted Gaussian functions centered at them. For any input $t$, we integrate these Gaussians over the fundamental box at $t$. This process maps persistence diagrams to smooth, absolutely integrable, Lipschitz functions. We proved that the sample mean distribution of Lipschitz continuous persistence curves (hence the Gaussian PCs) weakly converges to a Gaussian process. These curves proved successful and competitive with other TDA methods in the task of texture classification. The Gaussian PCs are one example of many summaries one can derive from the PC framework. The richness of PCs opens a door to several future directions of expansion for the theory around the framework such as bootstrapping, hypothesis testing, and stability analysis.

For any input $t$, we integrate these Gaussians over the fundamental box at $t$. This process maps persistence diagrams to smooth, absolutely integrable, Lipschitz functions. We proved that the sample mean distribution of Lipschitz continuous persistence curves (hence the Gaussian PCs) weakly converges to a Gaussian process. These curves proved successful and competitive with other TDA methods in the task of texture classification. The Gaussian PCs are one example of many summaries one can derive from the PC framework. The richness of PCs opens a door to several future directions of expansion for the theory around the framework such as bootstrapping, hypothesis testing, and stability analysis.


