

Supplemental Material for TopFusion

Audun Myers¹, Henry Kvinge^{1,2}, and Tegan Emerson^{1,3,4}

¹Pacific Northwest National Laboratory, Mathematics of Data Science

²University of Washington, Department of Mathematics

³Colorado State University, Department of Mathematics

⁴University of Texas El Paso, Department of Mathematical Sciences

1. Appendix

1.1. Sublevel Set Filtration

For times series data we assume a single variable function $f : \mathbb{R} \rightarrow \mathbb{R}$. Given $r \in \mathbb{R}$, we define the sublevel set below r as $f^{-1}(-\infty, r]$. As the filtration parameter r increases, we can tracking how the homology of the sublevel sets change. In the case of 1-D time series data these changes would only occur in D_0 since we are only interested in connected components and no changes would occur until reaching an extrema in the signal assuming the function f satisfies some general and standard conditions (e.g., q-tame [?]).

For extrema being local minima, we add new connected component “born” at height r_B . For the case of local extrema being maxima two existing sets (or components) are combined, each of which were born at r_B and r'_B . Form this information we follow the Elder Rule [?, p. 150], assuming $r_B \leq r'_B$ and the maxima is at r_D , then we say that the component born at r'_B dies going into r_D and the resulting set assumes the label r_B . The pair $(r'_B, r_D) \in D_0$ is called a persistence pair in the zero-dimensional sublevel set persistence. We continue generating the persistence pairs as r increase from $-\infty$ to ∞ . For the case of any unpaired births, we set a death coordinate of ∞ and label this persistence pair as an essential classes. As such, the resulting persistence diagram exists in the extended plane $\overline{\mathbb{R}^2}$.

The lifetime or persistence of a pair $(b_i, d_i) \in D_p(f)$ is defined as $\ell_i = d_i - b_i$. In this paper, our functions are only sampled on a finite domain, with the first sample at time t_a and the last sample at time t_b . We obtain a continuous function over $[t_a, t_b]$ by using a piecewise linear interpolation between consecutive samples, and extending the function to $\pm\infty$ by extending the first (resp., last) edges to rays. Doing so allows us to define a persistence diagram that does not have critical points on the boundary of our time series.

As such, we study the persistence points where both coordinates are finite, and omit persistence points that contain an unbounded coordinate. However, for demonstrative purposes we will still show the essential class persistence pair in our example in Fig. 1.

We now demonstrate an example signal in which we assume the time series extends as rays in the domain. This example function is shown in 1 having three local minima and two local maxima. The resulting persistence diagram

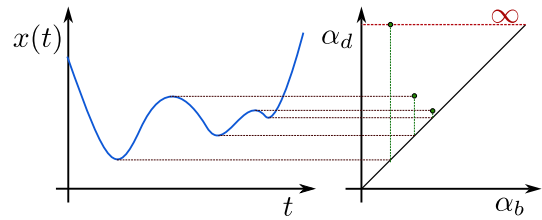


Figure 1. Demonstrative example of sublevel set persistence applied to time series data.

captures how these minima and maxima are paired together through sublevel set persistence with one essential class that will be ignored when we generate our persistence images. The algorithm used to calculate the this persistence diagram is detailed in [?].

The idea of sublevel set persistence can also be extended to higher dimensions. This is common in image data where our function is a surface. For images we can track the topology of the sublevel sets in both dimensions 0 and 1 as the sets can contain features in both dimensions.

1.2. Vietoris-Rips Filtration for Time Series

A simplicial complex is a generalization of a graph to higher dimensions, which are collections of simplices at various dimensions (e.g., points are zero-dimensional, edges are one-dimensional, and faces are two-dimensional simplices). These simplices are subsets of a vertex set

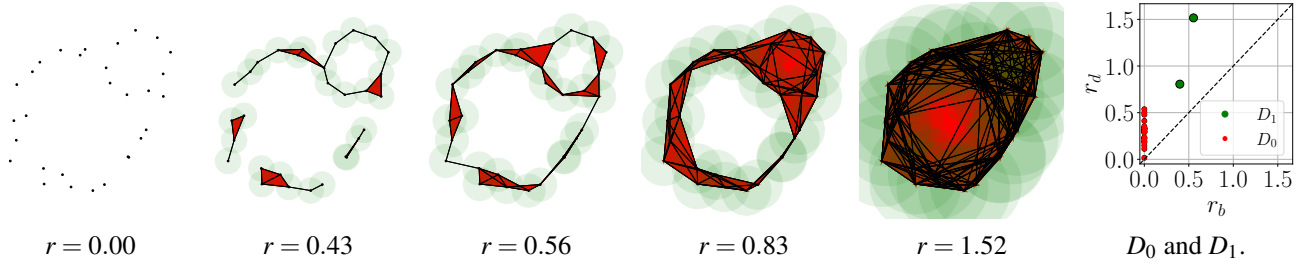


Figure 2. Example demonstrating persistent homology of point cloud data using the Vietoris-Rips complex filtration.

$\sigma \subset V$, and we require for face closure such that if $\sigma \in K$ and $\tau \subseteq \sigma$, then τ is also in K . For data that is stored as vectors (point cloud χ) we can directly form a simplicial complex using the distance between points. For time series data we can generate point cloud data by performing time-delay embedding. Time delay embedding of a time series embeds a signal $x(t) \in \mathbb{R}$ to $\chi(t) \in \mathbb{R}^n$ by using a delay τ with $\chi(t) = [x(t), x(t + \tau), x(t + 2\tau), \dots, x(t + (n - 1)\tau)]$. Applying this over the entire time series results in a point cloud χ .

To generate an abstract simplicial complex (ASC) from the point cloud, a Vietoris-Rips complex is used where we build the ASC K_r for any fixed parameter $r \geq 0$ by including all simplices with distance at most r ; i.e. $K_r = \{\sigma \subseteq V \mid d(u, v) \leq r \text{ for all } u, v \in \sigma\}$, where d is a distance function. Zero-dimensional simplices, the vertices of the complex, are all added at $r = 0$. An edge uv , which is a 1-dimensional simplex, is present in K_r for any r value above $d(u, v)$. Higher dimensional simplices such as triangles are included when all subedges are present.

Generating simplicial complexes as r increases results in a sequence of ASCs that we can use to study the persistent homology of the point cloud data. Persistent homology [?], a filtration tool from the field of Topological Data Analysis (TDA) [?, ?], is used to gain a sense of the shape and size of a dataset at multiple dimensions and filtration values. For example, it can measure connected components (dimension zero), holes (dimension one), voids (dimension two), and higher dimensional analogues, as well as an idea of their general size or geometry. Persistent homology measures these shapes using a parameterized filtration to detect when homology groups are born (appear) and die (disappear). In this case the parameterization filtration uses the parameter r for point cloud data.

By incrementing r we create a nested sequence of ASCs

$$K_0 \subseteq K_1 \subseteq K_2 \subseteq \dots \subseteq K_n. \quad (1)$$

We then calculate the homology of dimension p for each complex, $H_p(K_i)$, which is a vector space representing the p -dimensional structure of the space such as components, holes, voids, and higher dimensional features. However,

this information does not yet yield how the homology of each ASC is related to the next ASC. To get this information, persistent homology uses the inclusions on the ASCs to induce linear maps on the vector spaces resulting in a construction called a persistence module \mathcal{V} :

$$H_p(K_{\alpha_0}) \hookrightarrow H_p(K_{\alpha_1}) \hookrightarrow H_p(K_{\alpha_2}) \hookrightarrow \dots \hookrightarrow H_p(K_{\alpha_n}), \quad (2)$$

where \hookrightarrow are the maps induced by the inclusion map between ASCs. It should be noted that in the sequence of ASCs, each vertex must be unique and consistently identified.

The appearance and disappearance of classes at various dimensions in this object can be tracked, resulting in a summary known as a persistence diagram $\mathcal{D} = \{D_0, D_1, \dots, D_p\}$. For each homology generator which appears (born) at K_b and disappears (dies) at K_d , we add the persistence $[b, d]$ in the persistence diagram.

For example, consider Fig. 2 which shows point cloud data and the ASCs as $r \in [0.00, 0.43, 0.56, 0.83, 1.52]$. Additionally, this figure shows the corresponding persistence diagram for dimensions 0 and 1 as D_0 and D_1 , respectively.

For D_0 all persistence pairs are born at 0 since these are the components which begin as all the points in the point cloud represented as vertices in the first simplicial complex. These components begin to connect where persistence pairs in D_0 die following the Elder Rule. At $r = 0.56$, all of the components are combined into one component.

For D_1 the loops or holes in the ASCs are tracked through the filtration. At $r = 0.43$ the first, smaller hole is formed as shown in the ASC. This hole persists until approximately $r = 0.83$ in which it is filled resulting in the persistence pair $[0.43, 0.83]$. The second, larger hole first forms when the point cloud is fully connected at $r = 0.56$. This hole persists until $r = 1.52$ resulting in the persistence pair $[0.56, 1.52]$. We can see that the filtration of r results in two main persistence pairs in D_1 capturing the general shape and size of the data.

1.3. Persistence Images

Here we demonstrate how we calculate the persistence image from a persistence diagram. This procedure with a

toy example persistence diagram for calculating the PI for a persistence diagram D_p of dimension p is shown in Fig. 3.

The first step in the method is to get the birth-persistence representation of a persistence diagram $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, where each persistence pair $(b_i, d_i) \in D_p$ is mapped to the pair $(b_i, d_i - b_i)$ as shown in Fig. 3 (a). we will refer to the persistence (or commonly referred to as lifetime) of a persistence pair as $\ell_i = d_i - b_i$.

The second step is to represent $T(D_p)$ as a surface using each pair and a differentiable distribution function $g(x, y)$. Specifically, in this work we use the Gaussian kernel

$$g_G(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x-\mu_x)^2 + (y-\mu_y)^2 / 2\sigma^2}, \quad (3)$$

where σ is the user-chosen standard deviation parameter and μ_x and μ_y are the mean or center point of the distribution. In this work we set this parameter to $\sigma = 0.05p_{\max}$ for all experiments with p_{\max} as the maximum persistence. To define the surface function we also need a weighting function $w : \mathbb{R}^2 \rightarrow \mathbb{R}$ mapping persistence pairs to a weight. This weighting function must be continuous, differentiable, and intersect zero for a persistence pair with zero persistence. In this work we use the simple linear function

$$w(b_i, \ell_i) = \ell_i \quad (4)$$

We can now define our surface function of $T(D_p)$ as

$$S(x, y) = \sum_{(b_i, \ell_i) \in T(D_p)} w(b_i, \ell_i) g_G(x, y), \quad (5)$$

To practically apply the surface function as an impute to machine learning architectures we must create a finite sized representation of the surface. In this work we do this by creating a tessellation of squares (pixels) that cover the area occupied by

$$[b_{\min} - 3\sigma, b_{\min} + p_{\max} + 3\sigma] \times [-3\sigma, p_{\max} + 3\sigma]$$

resulting in a square area, where b_{\min} is the minimum birth value. These pixels have resolution $p \times p$ with the width of the square pixels w_p is chosen as $w_p = (p_{\max} + 6\sigma)/p$.

1.4. WESAD Embedding Parameters and Window Sizing

To determine a suitable window size we use the common criteria of at least $\ell = 20$ observations [?] and at most $\ell = 50$. In this work we consider an observation as one period of the signal. To do this we use the criteria of a period for a biological signal as 2τ , where τ is the time delay for delay embedding, as suggested in [?] for continuous time series. Additionally, we chose to use $\ell = 50$ as [?] found a general increase in performance with longer windows. As such, we set the window size as $w = 150\tau$, where we select

Table 1. Time delay embedding parameters for signals in WESAD dataset.

Location	Sensor	f_s (Hz)	Delay (τ)	Dim. (n)
Chest	ACC	700	65	5
Chest	ECG	700	38	4
Chest	EMG	700	5	4
Chest	EDA	700	4	4
Chest	Temp	700	3	4
Chest	Resp	700	600	3
Wrist	ACC	32	8	5
Wrist	BVP	64	16	4
Wrist	EDA	4	1	3
Wrist	Temp	4	1	4

τ using the mutual information method [?] due to the non-linear nature of the time series.

We also to shift the sliding windows by a distance $s = 0.25w_{\max}$ for each consecutive window, where w_{\max} is the largest window size of any of the signal types. This ended up being from the Resp signal as shown in Table 1 with $\tau = 600$. We only kept the window if the window fell within a region specified by only one state. Applying this procedure resulted in a total of 1620 windows across all subjects that were labeled as baseline, amused, or stressed.

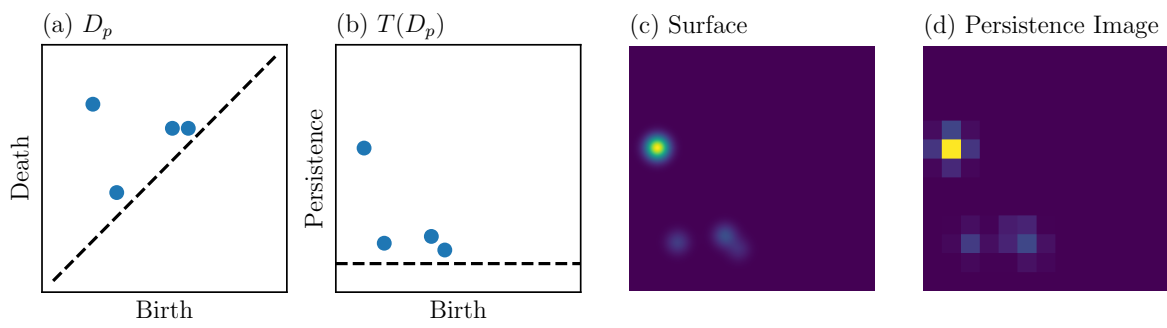


Figure 3. Persistence image pipeline.