Vectorizing Persistence Using Relative Homology

Benedikt Fluhr

Bielefeld University

4th Workshop on Computational Persistence

The Unravelled Relative Homology Lattice

- For a finite set X, let Filtr_X be the set of monotone maps $K_\bullet \colon [0,\infty) \to 2^X, \delta \mapsto K_\delta$ with $\{\{x\} \mid x \in X\} \subseteq K_0$ and K_δ a simplicial complex for any $\delta \geq 0$.
- Given a finite point cloud $X \subset \mathbb{R}^n$ we then have $\mathrm{Del}_\bullet(X) \in \mathrm{Filtr}_X$.
- For $K_\bullet \in \mathrm{Filtr}_X$ the associated *relative homology lattice* [Deh55] in degree $d \in \mathbb{N}$ is
$$\{(s,t) \in [0,\infty)^2 \mid s \leq t\} \to \mathrm{Vect}_{\mathbb{F}},\ (s,t) \mapsto H_d(K_t, K_s; \mathbb{F}),$$
- which is closely related to the *unravelled relative homology lattice*
$$h(K_\bullet) \colon \mathbb{M} \to \mathrm{Vect}_{\mathbb{F}},\ u \mapsto h(u; K_\bullet)$$
with $\mathbb{M} \subset \mathbb{R}^2$ a subposet shown in Fig. 1

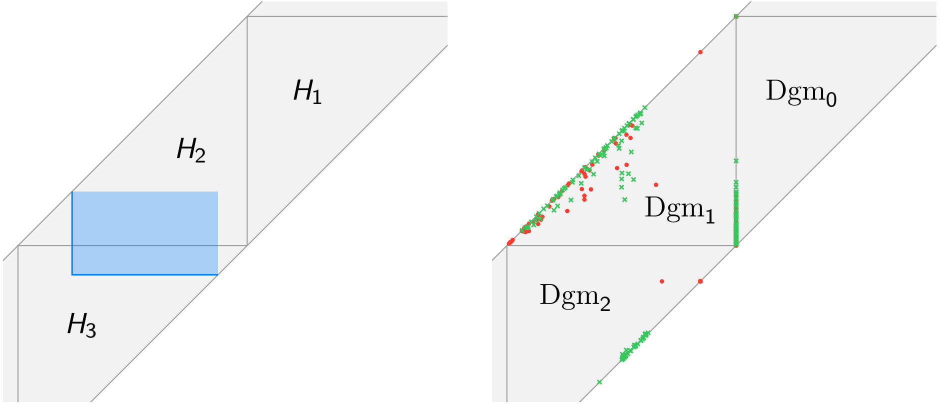

Figure 1: The poset $\mathbb{M}$ and the support for an indecomposable on the left. *Unravelled persistence diagrams* of two point clouds on the right; from a sphere (red disc) and a torus (green cross).

Vectorization as Hilbert Function

For a functor $F\colon \mathbb{M} \to \mathrm{Vect}_{\mathbb{F}}$ we have the *Hilbert function*

$$\mathrm{Hilb}(F)\colon \mathbb{M} \to \mathbb{R},\ u \mapsto \dim_{\mathbb{F}} F(u).$$

In summary:

- $\mathbb{R}^n \supset X \mapsto \mathrm{Del}_\bullet(X) \in \mathrm{Filtr}_X$
- $\mathrm{Filtr}_X \ni K_\bullet \mapsto h(K_\bullet) \in \mathrm{Vect}_{\mathbb{F}}^{\mathbb{M}}$
- $\mathrm{Vect}_{\mathbb{F}}^{\mathbb{M}} \ni F \mapsto \mathrm{Hilb}(F) \in \mathbb{R}^{\mathbb{M}}$

For a finite point cloud $X \subset \mathbb{R}^n$ we have $\mathrm{Hilb}(h(\mathrm{Del}_\bullet(X)) \in \mathcal{L}^2(\mathbb{M})$ and a factorization

$$\begin{array}{ccc} \mathrm{Del}_\bullet(X) & \longmapsto & \mathrm{Hilb}(h(\mathrm{Del}_\bullet(X))). \\ & \searrow \quad \nearrow & \\ & \mathrm{Dgm}_\bullet(\mathrm{Del}_\bullet(X)) & \end{array}$$

This way we obtain a *Hilbert kernel* on graded persistence diagrams [CEH07] implemented in `persunraveltorch`.

Results for the Hilbert Kernel

Binary classification of noisy point clouds from a sphere and a torus with subsampling:

- 10 point clouds for training each
- 250 points per point cloud with 50% of noise sampled uniformly from an enclosing cube
- 25 subsamples of 30 points each
- Accuracy: 95%
- Cross entropy: 0.3

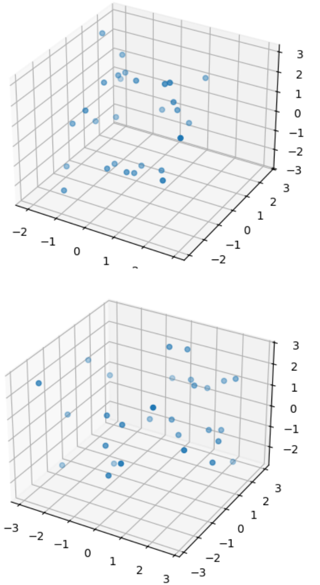

Figure 2: Subsamples of a noisy point cloud; from a sphere (top) and a torus (bottom).

Interpretation of Classifier

As the Hilbert kernel comes from an embedding into square-integrable functions $\mathbb{M} \to \mathbb{R}$, we have a straightforward interpretation of the SVC:

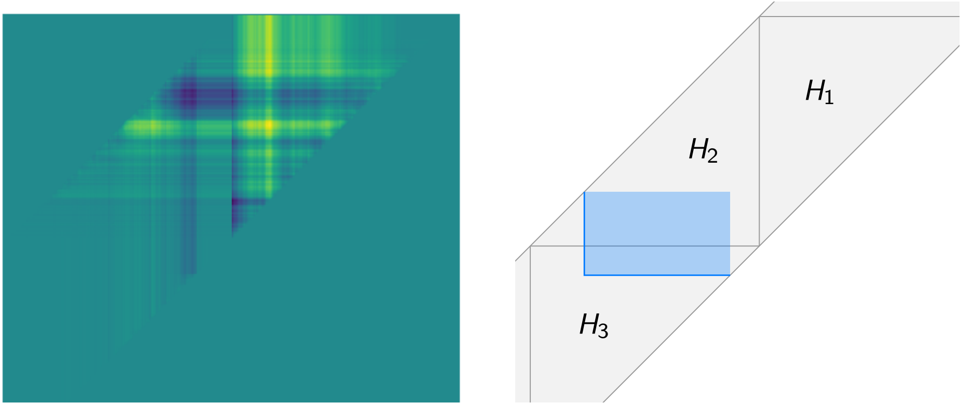

Figure 3: A rendering of the SVCs normal to the separating hyperplane (left) and the tessellation of $\mathbb{M}$ by homological degree (right).

Extension to Biplane

Given a function $f\colon \mathbb{M} \to \mathbb{R}$ we may define another function

$$\tilde{f}\colon \begin{cases} \{0,1\} \times \mathbb{M} & \to \mathbb{R}, \\ (0, \quad u) & \mapsto f(u), \\ (\quad 1, \quad u) & \mapsto f(\Sigma(u)) \end{cases}$$

on the *biplane* $\{0,1\} \times \mathbb{M}$, where $\Sigma\colon \mathbb{M} \to \mathbb{M}$ is a glide reflection corresponding to the degree-shift for the unravelled relative homology lattice.

Extension of $\tilde{f}\colon \{0,1\} \times \mathbb{M} \to \mathbb{R}$ by zero yields a function $\hat{f}\colon \{0,1\} \times \mathbb{R}^2 \to \mathbb{R}$.

We apply this to both, the Hilbert function of the unravelled relative homology lattice and the *unravelled rank invariant* inspired by [Wan+23] to obtain a "biplane bitmap" with two channels.

Biplane Cross-Correlation

We have a group action

$$
\begin{aligned}
(\mathbb{Z} \times \mathbb{R}^2) \times (\{0,1\} \times \mathbb{R}^2) &\to \{0,1\} \times \mathbb{R}^2, \\
((k,v),(d,u)) &\mapsto T^k(d, v+u),
\end{aligned}
$$

where

$$
\begin{aligned}
T\colon \{0,1\} \times \mathbb{R}^2 &\to \{0,1\} \times \mathbb{R}^2, \\
(0,u) &\mapsto (1,u), \\
(1,u) &\mapsto (0, u - (\mathrm{shift}, \mathrm{shift}))
\end{aligned}
$$

and shift is twice the width of $\mathbb{M}$.
For V a Euclidean vector space and compactly supported $\omega\colon \mathbb{Z} \times \mathbb{R}^2 \to V$ and $\hat{f}\colon \{0,1\} \times \mathbb{R}^2 \to \mathbb{R}$ we have the cross-correlation

$$
\omega * \hat{f}\colon \{0,1\} \times \mathbb{R}^2 \to V,\ p \mapsto \int_{\mathbb{Z}\times\mathbb{R}^2} \omega(g)\hat{f}(g.p)dg.
$$

Code for Biplane CNN

```
self.conv = nn.Sequential(
    ConvBiplane( 2, 4, (3, 3, 4), shift = pixel_columns ),
    nn.ReLU(),
    MaxPoolBiplane(2),
    ConvBiplane( 4, 8, (3, 3, 4), shift = pixel_columns // 2 ),
    nn.ReLU(),
    MaxPoolBiplane(2),
    nn.Flatten()
)

conv_out_features = self.conv( mock_biplane ).shape[1]

self.final_layer = nn.Linear( conv_out_features, nb_classes )
```

Results for Biplane CNN

Classification of noisy point clouds from a sphere, a torus, and a swiss roll with subsampling:

- 60 point clouds for training each
- 250 points per point cloud with 50% of noise sampled uniformly from an enclosing cube
- 80 subsamples of 30 points each
- Accuracy after 30 epochs: 100%
- Cross entropy after 150 epochs: 0.0015

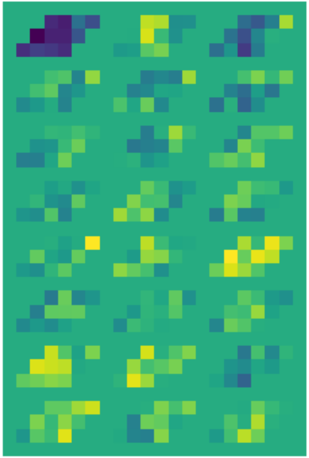

Figure 4: Filters of the first biplane convolutional layer as a pseudocolor image.

Inductive Biases

Definition (Flip-Invariance)

We say that a filter $\omega\colon \mathbb{Z} \times \mathbb{R}^2 \to V$ is *flip-invariant* if

$$\omega(k,(x,y)) = \omega(k,(y,x)) \quad \text{for all } k \in \mathbb{Z} \text{ and } x,y \in \mathbb{R}.$$

Now let $\Sigma\colon \mathbb{R}^2 \to \mathbb{R}^2$ be the glide reflection with $\Sigma(\mathbb{M}) = \mathbb{M}$ and corresponding to the degree-shift for the unravelled relative homology lattice.

Definition (Σ-Alternation)

We say that a function $g\colon \{0,1\} \times \mathbb{R}^2 \to V$ is Σ-*alternating* if $g|_{\{1\}\times\mathbb{R}^2} \equiv 0$ and

$$g(0,\Sigma(u)) = -g(0,u) \quad \text{for all } u \in \mathbb{R}^2.$$

Inclusion-Exclusion Principle

Theorem

Let $g\colon \{0,1\} \times \mathbb{R}^2 \to V$ be Σ-alternating and let $\omega\colon \mathbb{Z} \times \mathbb{R}^2 \to V$ be compactly supported and flip-invariant.
Let $K_\bullet, L_\bullet \in \mathrm{Filtr}_X$ and let $\hat{f}_0, \hat{f}_1, \hat{f}_2, \hat{f}_3\colon \{0,1\} \times \mathbb{R}^2 \to \mathbb{R}$ be the corresponding extensions of Hilbert functions of unravelled relative homology lattices of $K_\bullet \cap L_\bullet$, $K_\bullet$, $L_\bullet$, and $K_\bullet \cup L_\bullet$, respectively.
Then we have

$$\left\langle g, \omega * \hat{f}_3 \right\rangle_{\mathcal{L}^2} = \left\langle g, \omega * \hat{f}_1 \right\rangle_{\mathcal{L}^2} + \left\langle g, \omega * \hat{f}_2 \right\rangle_{\mathcal{L}^2} - \left\langle g, \omega * \hat{f}_0 \right\rangle_{\mathcal{L}^2}.$$

Here the upshot is, that by imposing inductive biases, removing activation functions and max pooling layers we obtain an invariant satisfying the inclusion-exclusion principle in the same way that removal of activation functions from an ordinary neural network yields an affine linear map.

API Documentation

https://persunraveltorch.neocities.org/latest/

[CEH07] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. "Stability of persistence diagrams". In: *Discrete Comput. Geom.* 37.1 (2007), pp. 103–120. ISSN: 0179-5376. DOI: 10.1007/s00454-006-1276-5.

[Deh55] René Deheuvels. "Topologie d'une fonctionnelle". In: *Ann. of Math. (2)* 61 (1955), pp. 13–72. ISSN: 0003-486X. DOI: 10.2307/1969619.

[Wan+23] Qiquan Wang et al. "Computable Stability for Persistence Rank Function Machine Learning". In: *arXiv e-prints* (July 2023). arXiv: 2307.02904 [math.AT].