



A Theory of Sub-Barcodes

Oliver A. Chubet  

Department of Computer Science, North Carolina State University, Raleigh, NC, USA

Kirk P. Gardner  

Department of Computer Science, North Carolina State University, Raleigh, NC, USA

Donald R. Sheehy  

Department of Computer Science, North Carolina State University, Raleigh, NC, USA

Abstract

The primary tool in topological data analysis (TDA) is persistent homology, which involves computing a *barcode* – often from point-cloud or scalar field data – that serves as a topological signature for the underlying function. In this work, we introduce *sub-barcodes* and show how they arise naturally from *factorizations* of persistence module homomorphisms. We show that, as a partial order induced by factorizations, the relation of being a sub-barcode is strictly stronger than the rank invariant, and we apply sub-barcode theory to the problem of inferring information about the barcode of an unknown Lipschitz function from samples. The advantage of this approach is that it permits strong guarantees – with no noise – while requiring no sampling assumptions, and the resulting barcode is guaranteed to be a sub-barcode of every Lipschitz function that agrees with the data. We also present an algorithmic theory that allows for the efficient approximation of sub-barcodes using filtered Delaunay triangulations for Euclidean inputs.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Mathematics of computing → Geometric topology; Theory of computation → Computational geometry; Computing methodologies → Algebraic algorithms

Keywords and phrases Topology, Topological Data Analysis, Persistent Homology, Persistence Modules, Barcodes, Sub-barcodes, Factorizations, Lipschitz Extensions

Digital Object Identifier 10.4230/LIPIcs.SoCG.2025.35

Related Version *Full Version*: <https://arxiv.org/abs/2206.10504> [10]

Funding This work was partially funded by the NSF under grant CCF-2017980.

1 Introduction

Topological data analysis (TDA) seeks to compute topological properties of unknown functions from finite samples. The most widely-used tool for TDA is persistent homology (PH), and the most widely-studied functions are distance functions. Thus, the algorithmic problems arising in this field tend to be either algebraic (PH computation), geometric (distance function representation), or both.

A PH pipeline starts with a real-valued function as input, and then considers the sublevel sets of this function. The homology of the sublevels at each threshold value are tracked and aggregated into a *barcode* (see Figure 1). The goal is to compute or approximate this barcode, which often requires substantial assumptions on the quality of the sample.

In this paper, we relax this goal to instead compute whatever *sub-barcode* is supported by the data we have. We form a sub-barcode by taking a sub-interval of each bar in the barcode. The formal definition of a sub-barcode and how it arises naturally in PH is covered in Section 3. The main idea is that, given *some* of the data, we compute *some* of the underlying structure.



© Oliver A. Chubet, Kirk P. Gardner, and Donald R. Sheehy;
licensed under Creative Commons License CC-BY 4.0

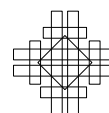
41st International Symposium on Computational Geometry (SoCG 2025).

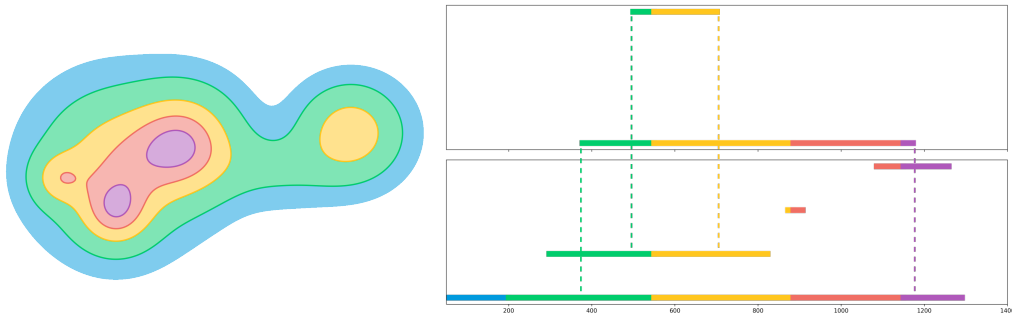
Editors: Oswin Aichholzer and Haitao Wang; Article No. 35; pp. 35:1–35:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** (Left) A contour plot of a Lipschitz function $f : X \rightarrow \mathbb{R}$. (Right) The barcode $\text{Bar}(f)$ of f (bottom) and a sub-barcode (top). All colors correspond to the function values indicated on the axis (bottom-right).

In practice, analyzing the topology of a finite sample entails two main challenges:

- **Extension:** If S is a finite sample of the underlying space X , then we need to extrapolate the values for a function f from the known values at S to the unknown values at the rest of X . We won't get the exact answer, but we would like some guarantees on the extension.
- **Discretization:** Having a good approximation to the function is insufficient: we also need a reliable discrete representation of the underlying space. In practice, these representations usually arise in the form of a simplicial complex or a simplicial set.

In the classic setting of distance functions in Euclidean space, strong sampling assumptions guarantee that the distance to a sample approximates the distance to the underlying set. The Delaunay triangulation provides a perfect discretization, capturing the exact homology of the distance sublevels. This was the original setting for PH [13].

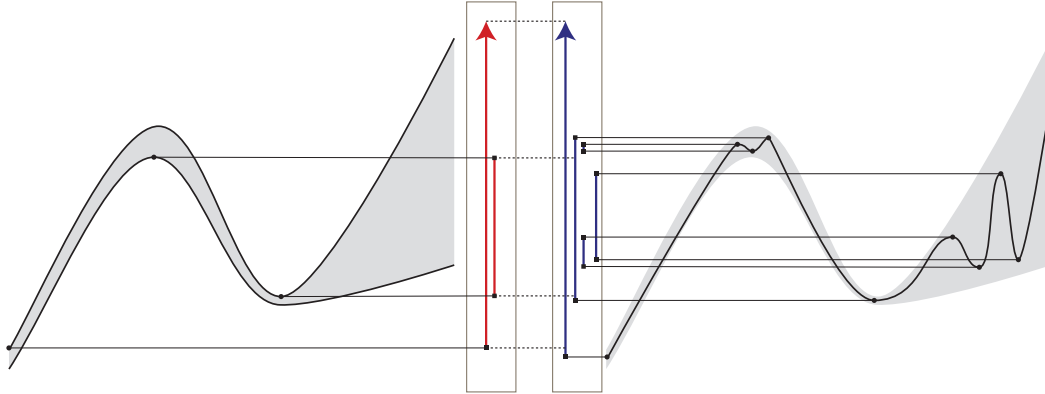
Lipschitz functions generalize distance functions and allow us to represent a much wider class of inputs. Several previous papers have shown that, given a sufficiently dense sample, a good approximation to the barcode of a Lipschitz function can be computed [6, 8, 11]. The guarantee from such work is that the output barcode is close to the true barcode in a standard metric called the bottleneck distance. Our approach eliminates all sampling assumptions, but gives guarantees in terms of a partial order on barcodes; that is, it is guaranteed to produce no noise or spurious bars, but finds only those topological features (bars) that are sufficiently supported by the sample.

Contributions and Outline

The background and definitions appear in sections 2 and 3. The *algebraic theory* of sub-barcodes is presented in Section 4. First, we show how sub-barcodes are related to the widely-studied rank invariant. Surprisingly, despite being a complete invariant for persistence, the natural extension of the rank invariant to a partial order is strictly less discriminating than sub-barcodes. This would be useless if it were not the case that sub-barcodes can be computed, a fact we show in the main sub-barcode theorem (Theorem 4), which states that sub-barcodes arise whenever we have one persistence module homomorphism factoring through another.

In Section 5, we lay out the *extension theory*. According to the algebraic theory of the previous section, the key to producing sub-barcodes is to find suitable upper and lower bounds on the unknown function. We show that maximum and minimum Lipschitz extensions serve

this purpose (Figure 2), as well as how to give good guarantees with only an approximation to the upper and lower bounds (Corollary 9). This section also contains a much more general result to encompass other classes of functions beyond Lipschitz functions (Theorem 11).



■ **Figure 2** On the left, two functions are depicted, one is an upper bound and the other is a lower bound on an unknown function $f : \mathbb{R} \rightarrow \mathbb{R}$. There is a corresponding barcode associated with the pair that matches minima in the upper bound to maxima in the lower bound. On the right is a candidate function f that lies between the upper and lower bounds and its barcode $\text{Bar}(f)$. The barcode of the inclusion of the upper and lower bounds is a sub-barcode of $\text{Bar}(f)$.

The *discretization theory* is presented in Section 6. Here, we show how to approximate the Lipschitz extension sub-barcode of the previous section using Delaunay triangulations. This leads to a theory of semi-supervised TDA in which only a subset of function values are known (Theorem 13). We also give a tighter approximation using the barycentric decomposition of the Delaunay triangulation (Theorem 14), and show that this method also applies to a wider class of functions (Example 15).

The *categorical theory* of sub-barcodes is presented in Section 7. The TDA and PH pipeline depend on category theory to express the way relationships between inputs impact relationships between outputs, i.e., barcodes. We relate sub-barcodes to the classic approach of smoothing barcodes and show how this arises naturally from categorical considerations. We also give a new perspective on barcodes in general for which sub-barcodes are shown to be true subobjects in the categorical sense, and discuss how this construction is related to the theory of fuzzy sets [15, 1, 21].

2 Background

Functions and Filtrations

Let (X, d) be a metric space. When the metric is Euclidean, we denote the distance as a norm, i.e., $d(x, y) := \|x - y\|$. The distance from a point x to a (finite) set U is denoted $d(x, U) := \min_{u \in U} d(x, u)$.

The real-valued functions $f : X \rightarrow \mathbb{R}$ form a vector space where $(f + \alpha g)(x) := f(x) + \alpha g(x)$, for all $\alpha \in \mathbb{R}$. We also have a poset of functions, where $f \geq g$ iff $f(x) \geq g(x)$ for all $x \in X$. A function $f : X \rightarrow \mathbb{R}$ is λ -Lipschitz if, for all $x, y \in X$, we have

$$f(y) - \lambda d(x, y) \leq f(x) \leq f(y) + \lambda d(x, y).$$

If λ is not specified, then it is assumed that $\lambda = 1$. This is perhaps not the most common way to express the Lipschitz condition, but it is closest to how it is used in this work.

The t -sublevel set $\text{Sub}_f(t)$ of f is the set of points $x \in X$ with $f(x) \leq t$:

$$\text{Sub}_f(t) := f^{-1}((-\infty, t]) = \{x \in X \mid f(x) \leq t\}.$$

This notation is designed to indicate that Sub_f is a *filtration*: for $s \leq t$ we have $\text{Sub}_f(s) \subseteq \text{Sub}_f(t)$. There is a partial order on sublevel filtrations where $\text{Sub}_f \subseteq \text{Sub}_g$ iff $\text{Sub}_f(t) \subseteq \text{Sub}_g(t)$ for all $t \in \mathbb{R}$. It is a simple exercise to check that, if $f \geq g$, then $\text{Sub}_f \subseteq \text{Sub}_g$; i.e., larger functions have smaller sublevel sets.

► **Remark 1.** The mapping from a function to its sublevel filtration is a contravariant functor from the poset of functions to the poset of filtrations. The filtrations themselves are functors $\mathbb{R} \rightarrow \mathbf{Top}$ from the totally ordered set \mathbb{R} of real numbers to \mathbf{Top} , the category of topological spaces. Throughout, we use remarks like this one to highlight the categorical interpretations. The reader who is unfamiliar with these terms may still follow all definitions and algorithms.

Persistence Modules

A *pointwise finite dimensional (p.f.d.) persistence module* $M : \mathbb{R} \rightarrow \mathbf{vec}$ assigns a (finite-dimensional) vector space $M(t)$ to all $t \in \mathbb{R}$ and a linear map $M_{s \leq t} : M(s) \rightarrow M(t)$ for all $s \leq t$. The map $M_{s \leq s}$ is the identity on $M(s)$, and for all $s \leq t \leq u$, $M_{s \leq u} = M_{t \leq u} \circ M_{s \leq t}$. For persistence modules M and N , a homomorphism $\phi : M \rightarrow N$ assigns linear maps $\phi_t : M(t) \rightarrow N(t)$ for all $t \in \mathbb{R}$ so that for all $s \leq t$, we have $\phi_t \circ M_{s \leq t} = N_{s \leq t} \circ \phi_s$.

► **Remark 2.** A persistence module M is a functor $M : \mathbb{R} \rightarrow \mathbf{vec}$, and a persistence module homomorphism $M \rightarrow N$ is a natural transformation $\phi : M \Rightarrow N$ between functors $\mathbb{R} \rightarrow \mathbf{vec}$, so that the category of persistence modules is precisely the category $\mathbf{vec}^{\mathbb{R}} = [\mathbb{R}, \mathbf{vec}]$ of functors $\mathbb{R} \rightarrow \mathbf{vec}$.

A *factorization* of homomorphisms $F : M \rightarrow N$ to $G : M' \rightarrow N'$ is given by a pair of homomorphisms $\varphi = (\varphi_1 : M' \rightarrow M, \varphi_2 : N \rightarrow N')$ such that $G = \varphi_2 \circ F \circ \varphi_1$ (Diagram 1).

$$\begin{array}{ccc} M & \xleftarrow{\varphi_1} & M' \\ \downarrow F & & \downarrow G \\ N & \xrightarrow{\varphi_2} & N' \end{array} \quad (1)$$

Let $\mathbf{Fac}(\mathbf{vec}^{\mathbb{R}})$ denote the category with homomorphisms of persistence modules as objects, and arrows $\varphi : F \rightarrow G$. Then $\varphi : F \rightarrow G$ is a factorization of G through F .

► **Example 3.** Letting $M(\epsilon)$ denote a persistence module M shifted by a constant $\epsilon \in \mathbb{R}$, a 2ϵ -interleaving is traditionally given by a pair of commuting diagrams

$$\begin{array}{ccc} M(-\epsilon) & \xrightarrow{M_\epsilon} & M(\epsilon) \\ & \searrow \Phi(-\epsilon) & \nearrow \Psi \\ & N & \end{array} \quad (2) \quad \begin{array}{ccc} & & M \\ \Psi(-\epsilon) \nearrow & & \searrow \Phi \\ N(-\epsilon) & \xrightarrow{N_\epsilon} & N(\epsilon) \end{array} \quad (3)$$

depicting factorizations of homomorphisms $M_\epsilon : M(-\epsilon) \rightarrow M(\epsilon)$ and $N_\epsilon : N(-\epsilon) \rightarrow N(\epsilon)$ by a pair $(\Phi : M \rightarrow N(\epsilon), \Psi : N \rightarrow M(\epsilon))$. In this section, we show that the commutativity of Diagrams (2) and (3) imply sub-barcode relations $\text{Bar}(M_\epsilon) \subseteq \text{Bar}(N)$ and $\text{Bar}(M) \supseteq \text{Bar}(N_\epsilon)$.

Homology

Homology is a topological invariant that is particularly useful because it is both informative and efficiently computable for triangulable topological spaces. A homology functor $H : \mathbf{Top} \rightarrow \mathbf{vec}$ associates a vector space $H(X)$ to a topological space X , and takes continuous maps between topological spaces to linear maps between the corresponding vector spaces.

Persistence modules naturally arise in TDA by taking the homology of a sublevel filtration of a function. This assembles the functorial TDA pipeline from functions to filtrations to persistence modules. It is useful to see these as functors and not just functions because the additional structure of functoriality tells us that, if $f \geq g$, then there is a persistence module homomorphism $H \text{Sub}_f \rightarrow H \text{Sub}_g$. As we will see, these homomorphisms can be combined into factorizations (leading to sub-barcodes) and interleavings (leading to approximations).

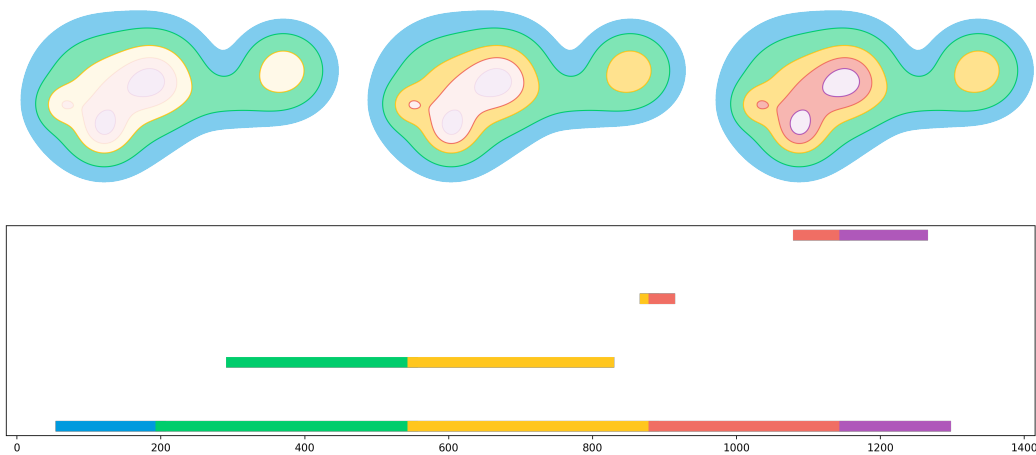


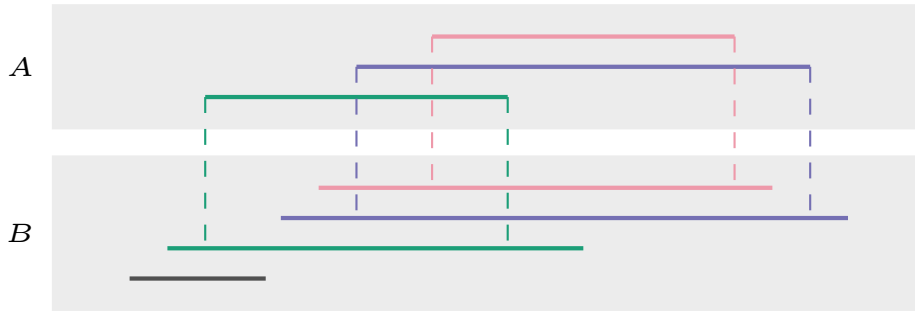
Figure 3 Sublevels of a 3-Lipschitz function $f : X \rightarrow \mathbb{R}$ on a subset of \mathbb{R}^2 (top) and its barcode $\text{Bar}(f) = \text{Bar}(H_1 \text{Sub}_f)$ in dimension 1 (bottom). Colors correspond to the function values indicated on the bottom-axis.

3 Barcodes and Sub-Barcodes

Let \mathcal{I} denote the poset of intervals on \mathbb{R} ordered by inclusion. A *barcode* B is a function $B : \underline{B} \rightarrow \mathcal{I}$ where \underline{B} is a set whose elements $\beta \in \underline{B}$ are called *bars*. The barcode assigns an interval to each bar. Note that B is not necessarily injective. When there is no confusion, we identify the bars in a barcode with their corresponding intervals. A *sub-barcode mapping* between barcodes A and B is a function $\phi : \underline{A} \rightarrow \underline{B}$ such that, for all $\alpha \in \underline{A}$, we have $A(\alpha) \subseteq B(\phi(\alpha))$. If ϕ is injective, then we call it a *sub-barcode matching* and say A is a *sub-barcode* of B , denoted $A \sqsubseteq B$. That is, A is a sub-barcode of B if A can be formed from B by taking a subinterval of each bar in a subset of \underline{B} (Figure 4).

There is a barcode $\text{Bar}(M)$ corresponding to any persistence module M . Note, the image of a persistence module homomorphism $\phi : M \rightarrow N$ is also persistence module. We overload notation and write $\text{Bar}(\phi)$ to indicate $\text{Bar}(\text{im } \phi)$.

From the PH pipeline, we have persistence modules $H \text{Sub}_f$ associated with each continuous function $f : X \rightarrow \mathbb{R}$. We again simplify our notation and write $\text{Bar}(f)$ to denote $\text{Bar}(H \text{Sub}_f)$ and $\text{Bar}(f \geq g)$ to denote $\text{Bar}(\text{im } H(\text{Sub}_f \subseteq \text{Sub}_g))$. The object giving us a barcode is always clear from context.



■ **Figure 4** Two barcodes A and B and a sub-barcode matching $M : A \rightarrow B$.

We now have enough definitions to state the main theorem of sub-barcodes. The proof is postponed until Section 4 when we will have developed some necessary tools.

► **Theorem 4 (The Sub-Barcode Theorem).** *If there exists a factorization $\varphi : F \rightarrow G$ of persistence module homomorphisms (i.e., $G = \varphi_2 F \varphi_1$), then $\text{Bar}(G) \sqsubseteq \text{Bar}(F)$.*

4 The Algebraic Theory

4.1 Sub-barcodes vs. Ranks

Let $\text{rk } m$ denote the *rank* of a linear map m defined as the dimension of its image, $\text{im } m$. The *rank invariant* rank_M of a persistence module $M : \mathbb{R} \rightarrow \mathbf{vec}$ is a function from ordered pairs in \mathbb{R} to the natural numbers defined as $\text{rank}_M(s \leq t) := \text{rk } M_{s \leq t}$. It is straightforward to extend this invariant to homomorphisms $F : M \rightarrow N$ by letting $\text{rank}_F(s \leq t) := \text{rk}(F_t \circ M_{s \leq t})$. In the following, we consider the rank invariant for a persistence module M to be the rank invariant of its identity homomorphism, i.e., $\text{rank}_M := \text{rank}_{1_M}$.

A basic fact from linear algebra is that, if a linear map g factors through a second linear map f , then $\text{rk}(g) \leq \text{rk}(f)$. It follows immediately that, if there is a factorization $F \rightarrow G$ of persistence module homomorphisms, then for all $s \leq t$,

$$\text{rank}_G(s \leq t) \leq \text{rank}_F(s \leq t).$$

This puts a partial order on rank invariants that we call the *sub-rank relation* and denote by $\text{rank}_G \leq \text{rank}_F$.

► **Remark 5.** rk is a contravariant functor $\mathbf{Fac}(\mathbf{vec}) \rightarrow \mathbb{N}$ and rank is a contravariant functor $\mathbf{Fac}(\mathbf{vec}^{\mathbb{R}}) \rightarrow \mathbb{N}$, i.e., from factorizations of vector spaces (resp. persistence modules) to the poset of natural numbers. An ordering of two rank invariants is a natural transformation between these functors.

In light of the sub-barcode theorem, the rank invariant is the same *type* of invariant as sub-barcodes. It puts a partial order on morphisms induced by factorizations. However, although the barcode can be constructed from the rank invariant, the natural ordering of rank invariants is a weaker invariant than sub-barcodes as expressed in the following.

► **Theorem 6.** *The sub-barcode order is a more discriminating invariant than the sub-rank order in the sense that, for all persistence module homomorphisms F and G , $\text{Bar}(G) \sqsubseteq \text{Bar}(F)$ implies that $\text{rank}_G \leq \text{rank}_F$, but not vice-versa.*

Proof. The rank invariant is easily extracted from a barcode: $\text{rank}_F(s \leq t)$ is the number of bars in $\text{Bar}(F)$ that contain the closed interval $s \leq t$. It follows immediately from the injectivity of the sub-barcode matching (and the pigeonhole principle) that $\text{Bar}(G) \sqsubseteq \text{Bar}(F)$ implies that $\text{rank}_G \leq \text{rank}_F$.

For the other side of the proof, it suffices to give an example of persistence modules M and N such that $\text{rank}_M \leq \text{rank}_N$ while $\text{Bar}(M) \not\sqsubseteq \text{Bar}(N)$.¹ Let M be a persistence module where $\text{Bar}(M)$ has two bars corresponding to intervals $[0, 1)$ and $[2, 3)$, and let N be a persistence module such that $\text{Bar}(N)$ has only one bar corresponding to the interval $[0, 3)$. Clearly, $\text{rank}_M \leq \text{rank}_N$, but it is not possible for a barcode with two bars to be a sub-barcode of one with only one bar. Thus $\text{Bar}(M) \not\sqsubseteq \text{Bar}(N)$. ◀

4.2 Induced Matching Theory Proof of the Sub-Barcode Theorem

Although not expressed in these terms, Bauer and Lesnick’s theory of induced matchings [3] shows that sub-barcode matchings arise from certain homomorphisms. The induced matchings in that work are matchings between the bars that do not (necessarily) satisfy the inclusion requirement of a sub-barcode matching. They showed that for any homomorphism $F : M \rightarrow N$, there is a partial bijective function (a *matching*) between the bars with the property that every bar of $\text{Bar}(F)$ is the intersection of the intervals of the matched bars from $\text{Bar}(M)$ to $\text{Bar}(N)$. These are constructed from the pair of canonical (functorial) matchings $\text{Bar}(M) \rightarrow \text{Bar}(F) \rightarrow \text{Bar}(N)$ induced by the epi-mono factorization of F through its image. Translated into the vocabulary of sub-barcodes, this result implies that $\text{Bar}(F) \sqsubseteq \text{Bar}(M)$ and $\text{Bar}(F) \sqsubseteq \text{Bar}(N)$.

More generally, if F is a monomorphism (all maps F_t are injective), then $\text{Bar}(M) \sqsubseteq \text{Bar}(N)$. This is because the induced matching of a monomorphism matches every bar, in addition to the property that, if $[a, b]$ is matched to $[c, d]$, then $a \geq c$ and $b = d$; thus, a submodule relation between persistence modules implies a sub-barcode relation between the barcodes.

If F is an epimorphism (all maps F_t are surjective), then the induced matching is surjective, and if $[a, b]$ is matched to $[c, d]$, then $a = c$ and $d \leq b$. So, reversing this induced matching, we get $\text{Bar}(N) \sqsubseteq \text{Bar}(M)$. We use these matchings in the proof of the Sub-Barcode Theorem below.

Proof of The Sub-Barcode Theorem (Theorem 4). The factorization $\varphi = (\varphi_1, \varphi_2) : F \rightarrow G$ entails $G = \varphi_2 F \varphi_1$. Let $m : \text{im } G \hookrightarrow \text{im } \varphi_2 F$ be the unique monomorphism given by the universal property of images, and let $e : \text{im } F \rightarrow \text{im } \varphi_2 F$ be the epimorphism given by restricting φ_2 to $\text{im } F$. Using the induced matching theory, m gives a sub-barcode matching $\text{Bar}(\varphi_2 F) \sqsubseteq \text{Bar}(F)$. Then, the reverse of the matching induced by e gives a sub-barcode matching $\text{Bar}(G) \sqsubseteq \text{Bar}(\varphi_2 F)$. Putting these together, we get

$$\text{Bar}(G) \sqsubseteq \text{Bar}(\varphi_2 F) \sqsubseteq \text{Bar}(F). \quad \blacktriangleleft$$

The following corollary for ordered sequences of functions forms the basis for all of our applications of sub-barcodes in TDA.

► **Corollary 7.** *If $h \geq g \geq f \geq e$ are functions $X \rightarrow \mathbb{R}$, then $\text{Bar}(h \geq e) \sqsubseteq \text{Bar}(g \geq f)$.*

Proof. It suffices to observe that the ordering on the functions gives a factorization $(g \geq f) \rightarrow (h \geq e)$ of $h \geq e$ through $g \geq f$. By the functoriality of the sublevel functor and homology, we get the corresponding factorization of persistence modules. The claim then follows from Theorem 4. ◀

¹ As before, rank_M is shorthand for rank_{1_M} .

5 The Extension Theory

Given an unknown Lipschitz function $f : X \rightarrow \mathbb{R}$ and a sample $S \subset X$, let f_S denote the function f restricted to the points of S . The pair (S, f_S) denotes input to a TDA problem where the goal is to infer as much as possible about f . Any function $g : X \rightarrow \mathbb{R}$ that agrees with f_S at the points of S is called an *extension* of f_S . If g is also Lipschitz, then it is called a *Lipschitz extension*. By the definition of Lipschitz continuity, each point $s \in S$ puts an upper bound and a lower bound on the unknown f so that

$$f(s) - d(x, s) \leq f(x) \leq f(s) + d(x, s).$$

The combination of these upper and lower bounds gives the *maximum and minimum Lipschitz extensions*, defined respectively for each $x \in X$ as

$$\check{f}_S(x) := \min_{s \in S} f(s) + d(x, s) \quad \text{and} \quad \hat{f}_S(x) := \max_{s \in S} f(s) - d(x, s).$$

Note that the max extension is the minimum of the upper bounds and vice versa (see also [17]).

The following theorem states that, for every Lipschitz function f that agrees with the data, the max and min Lipschitz extensions yield a barcode that is contained in $\text{Bar}(f)$.

► **Theorem 8.** *Given $S \subseteq X$ and $f_S : S \rightarrow \mathbb{R}$, for all Lipschitz extensions f of f_S , we have $\text{Bar}(\check{f}_S \geq \hat{f}_S) \subseteq \text{Bar}(f)$.*

Proof. It suffices to observe that for all Lipschitz extensions of f , we have $\check{f}_S \geq f \geq \hat{f}_S$. Because $\text{Bar}(f) = \text{Bar}(f \geq f)$, the result follows from Corollary 7. ◀

The Choice of Lipschitz Constants

It is natural to ask whether (and how) the choice of the Lipschitz constant affects this result. The previous statements have all assumed 1-Lipschitz functions. More generally, tuning the Lipschitz constant naturally leads to a filtration of the barcode itself as shown in the following.

► **Corollary 9.** *For any $t \in \mathbb{R}$, let $\check{f}_S^{(t)}$ and $\hat{f}_S^{(t)}$ denote the (resp.) max and min t -Lipschitz extensions. If f_S is t_0 -Lipschitz and $t_0 \leq t_1 \leq t_2$, then $\text{Bar}(\check{f}_S^{(t_1)} \geq \hat{f}_S^{(t_1)}) \subseteq \text{Bar}(\check{f}_S^{(t_2)} \geq \hat{f}_S^{(t_2)})$.*

A Sub-Barcode that is Close in Bottleneck Distance

A fundamental challenge in computing sub-barcodes from Lipschitz extensions is that it is not clear how to construct a simplicial complex that can represent the sublevel sets of both the max and min extension at the same time. So, in order to still derive guarantees, it will suffice to have approximations to these extensions. The following theorem shows that a bottleneck-close sub-barcode is possible if we have close approximations to the upper and lower bounds.

► **Theorem 10.** *Let $u' \geq u \geq f \geq \ell \geq \ell'$ be an ordered sequence of functions $X \rightarrow \mathbb{R}$ such that $\|\ell - \ell'\|_\infty \leq \varepsilon$ and $\|u - u'\| \leq \varepsilon$. Then, $d_B(\text{Bar}(u \geq \ell), \text{Bar}(u' \geq \ell')) \leq \varepsilon$.*

Proof. The proof is a straightforward application of the general stability theorem for image persistence. The bounds $\|\ell - \ell'\|_\infty \leq \varepsilon$ and $\|u - u'\| \leq \varepsilon$ imply that $\ell' + \varepsilon \geq \ell \geq \ell'$ and $u' \geq u \geq u' - \varepsilon$. As all maps from ordered functions induce inclusions of sublevel sets, all required maps commute to form an interleaving in the arrow category of filtrations. This gives an interleaving in the arrow category of persistence modules which leads to an ε -interleaving of the image persistence modules. ◀

Beyond Lipschitz Functions

The careful reader may note that there is nothing particularly special about Lipschitz functions in the treatment above. One could consider other classes of functions as well. More generally, it suffices to have an upper and lower bound on the unknown function. In the Lipschitz case, the Lipschitz extensions form a lattice of hypotheses that we are considering. Recall that a poset is a lattice if every subset has a join (least upper bound) and a meet (greatest lower bound). The symbol \vee is used for joins and \wedge is used for meets. This is the origin of our notation \check{f}_S and \hat{f}_S .

For a set \mathcal{F} of functions, let $\vee \mathcal{F} := \bigvee_{f \in \mathcal{F}} f$ and $\wedge \mathcal{F} := \bigwedge_{f \in \mathcal{F}} f$. In particular, if \mathcal{F} denotes the set of Lipschitz extensions of f_S , then $\check{f}_S = \vee \mathcal{F}$ and $\hat{f}_S = \wedge \mathcal{F}$. For other lattices of hypotheses, the same proof as for the Lipschitz case (Theorem 8) implies the following theorem.

► **Theorem 11.** *Given a lattice \mathcal{F} of functions $X \rightarrow \mathbb{R}$, for all $f \in \mathcal{F}$, we have*

$$\text{Bar}(\bigvee \mathcal{F} \geq \bigwedge \mathcal{F}) \sqsubseteq \text{Bar}(f).$$

6 The Discretization Theory

The preceding sections laid out the theory of sub-barcodes for persistence modules, as well as the useful setting of Lipschitz functions. The following section shows how to approximate these barcodes.

The fundamental challenge is to provide a single discretization of space that approximates (or at least bounds) both Lipschitz extensions at the same time. We assume throughout this section that X is a convex polytope in \mathbb{R}^d .² For a finite set $S \subset X$, the *Voronoi diagram* assigns a convex polytope called the *Voronoi cell* to each point $s \in S$ defined as

$$\text{Vor}_S(s) := \{x \in X \mid d(x, s) = d(x, S)\}.$$

The *radius* of $\text{Vor}_S(s)$ is the maximum distance from s to any point in the cell: $r_s := \max_{x \in \text{Vor}_S(s)} d(x, s)$. More generally, for $\sigma \subseteq S$, we have a Voronoi cell

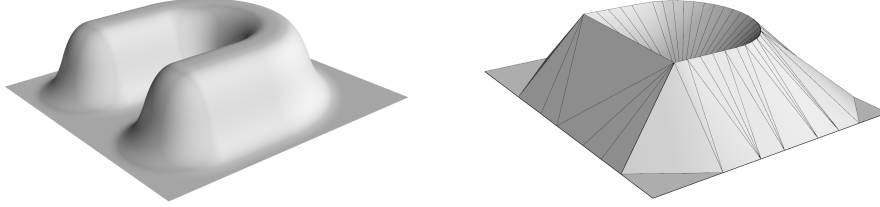
$$\text{Vor}_S(\sigma) := \bigcap_{s \in \sigma} \text{Vor}_S(s).$$

Dually, the *Delaunay triangulation* Del_S associated with S is the simplicial complex with vertex set S and simplices $\text{Del}_S = \{\sigma \subseteq S \mid \text{Vor}_S(\sigma) \neq \emptyset\}$. The Delaunay triangulation is widely used for persistent homology on Euclidean inputs.

6.1 Simple Voronoi Extensions and a Delaunay Filtration

The most direct way to use a Delaunay triangulation to estimate the persistence diagram of a set of points is to extend the function from the vertex set to the simplices, setting the value at the simplex be the maximum value at its vertices. This corresponds to the piecewise linear extension of the function to the geometric realization of the complex (see Section 2.5, Morozov [20]). Unfortunately, this can fail badly in the absence of strong sampling

² The convexity of X can be relaxed in many settings. Assuming X is convex simplifies our use of the Nerve theorem.



■ **Figure 5** A function and its linear extension onto the Delaunay triangulation of a sample. The discretization introduces a spurious feature because it does not register the opening.

guarantees; Figure 5 shows an example where the piecewise linear extension introduces spurious features. Thus, the problem with some samples is not only that we can miss features, but that we may hallucinate them as well.

The key to using a discretization is to use an image filtration (see Cohen-Steiner et al. [12], Bauer [4]) capturing both an upper and lower bound. The following theorem states that we can compute a sub-barcode for any Lipschitz function by filtering the Delaunay triangulation by upper and lower bounds given on the input values and the radius of the Voronoi cells.

► **Theorem 12.** *Let S be a finite subset of a polytope $X \subset \mathbb{R}^d$. Let $f_S : S \rightarrow \mathbb{R}$ be a sample of an unknown Lipschitz function $f : X \rightarrow \mathbb{R}$. For each $s \in S$, let r_s denote the radius of the Voronoi cell $\text{Vor}_S(s)$. Let u and ℓ be functions $\text{Del}_S \rightarrow \mathbb{R}$ defined on Delaunay simplices $\sigma \in \text{Del}_S$ as*

$$u(\sigma) := \max_{s \in \sigma} f_S(s) + r_s \quad \text{and} \quad \ell(\sigma) := \max_{s \in \sigma} f_S(s) - r_s.$$

Then, $\text{Bar}(u \geq \ell) \sqsubseteq \text{Bar}(f)$.

Proof. To relate the barcodes, we will first construct functions on X that have the same barcode as the simplicial functions u and ℓ . Second, we will relate these functions to the Lipschitz extensions \check{f}_S and \hat{f}_S .

We define piecewise constant functions on X using the Voronoi cells and f_S as

$$u^*(x) := \max_{s | x \in \text{Vor}_S(s)} f_S(s) + r_s \quad \text{and} \quad \ell^*(x) := \max_{s | x \in \text{Vor}_S(s)} f_S(s) - r_s.$$

By the Persistent Nerve Lemma (Chazal et al. [9], Lemma 3.4), there is a homotopy equivalence between the sublevels of u and u^* that commutes with inclusions (see also [2]). The same holds for ℓ and ℓ^* . It follows that $\text{Bar}(u \geq \ell) \cong \text{Bar}(u^* \geq \ell^*)$.

For any $x \in X$, there is an $s \in S$ such that $x \in \text{Vor}_S(s)$ and $u^*(x) = f_S(s) + r_s$, so

$$u^*(x) = f_S(s) + r_s \geq f_S(s) + d(x, s) \geq \min_{s_0 \in S} f_S(s_0) + d(x, s_0) = \check{f}_S(x).$$

Similarly, there is an $s \in S$ such that $x \in \text{Vor}_S(s)$ and $\ell^*(x) = f_S(s) - r_s$, so

$$\ell^*(x) = f_S(s) - r_s \leq f_S(s) - d(x, s) \leq \max_{s_0 \in S} f_S(s_0) - d(x, s_0) = \hat{f}_S(x),$$

thus $u^* \geq \check{f}_S \geq \hat{f}_S \geq \ell^*$, and Corollary 7 therefore implies

$$\text{Bar}(u \geq \ell) \cong \text{Bar}(u^* \geq \ell^*) \sqsubseteq \text{Bar}(\check{f}_S \geq \hat{f}_S) \sqsubseteq \text{Bar}(f). \quad \blacktriangleleft$$

6.2 Semi-Supervised TDA

We have assumed thus far that we have access to function values at all points in the sample. This resembles a supervised learning problem. If instead we only have function values at a subset $P \subset S$, we can still use the points of S to improve our approximation. We call this *semi-supervised TDA*.

The most common guarantees in PH are derived from interleavings, and thus yield bottleneck distance bounds on the resulting barcodes. In this section, we combine those results with the Lipschitz extension sub-barcode and the Delaunay filtration above to get a guaranteed approximation. Throughout, we assume that S is a sample of X , and that we only have function values $f_P : P \rightarrow \mathbb{R}$ for $P \subset S$. The algorithm extends \check{f}_P and \hat{f}_P to the points of S and then use the Delaunay filtration from the previous subsection.

In the following, we say that S is an ε -sample of X if every point of X is within ε of a point in S . Equivalently, the radius of every Voronoi cell $\text{Vor}_S(s)$ is at most ε .

► **Theorem 13.** *Let $X \subset \mathbb{R}^d$ be a convex polytope, and suppose $S \subset X$ is an ε -sample. Let $f_P : P \rightarrow \mathbb{R}$ be a sample of an unknown Lipschitz function $f : X \rightarrow \mathbb{R}$ defined on a subset $P \subset S$, and let u and ℓ be functions $\text{Del}_S \rightarrow \mathbb{R}$ defined on each Delaunay simplex $\sigma \subseteq S$ as*

$$u(\sigma) = \max_{s \in \sigma} \check{f}_P(s) + \varepsilon \quad \text{and} \quad \ell(\sigma) = \max_{s \in \sigma} \hat{f}_P(s) - \varepsilon.$$

Then, $\text{Bar}(u \geq \ell) \subseteq \text{Bar}(f)$, and $d_B(\text{Bar}(u \geq \ell), \text{Bar}(\check{f}_P \geq \hat{f}_P)) \leq \varepsilon$.

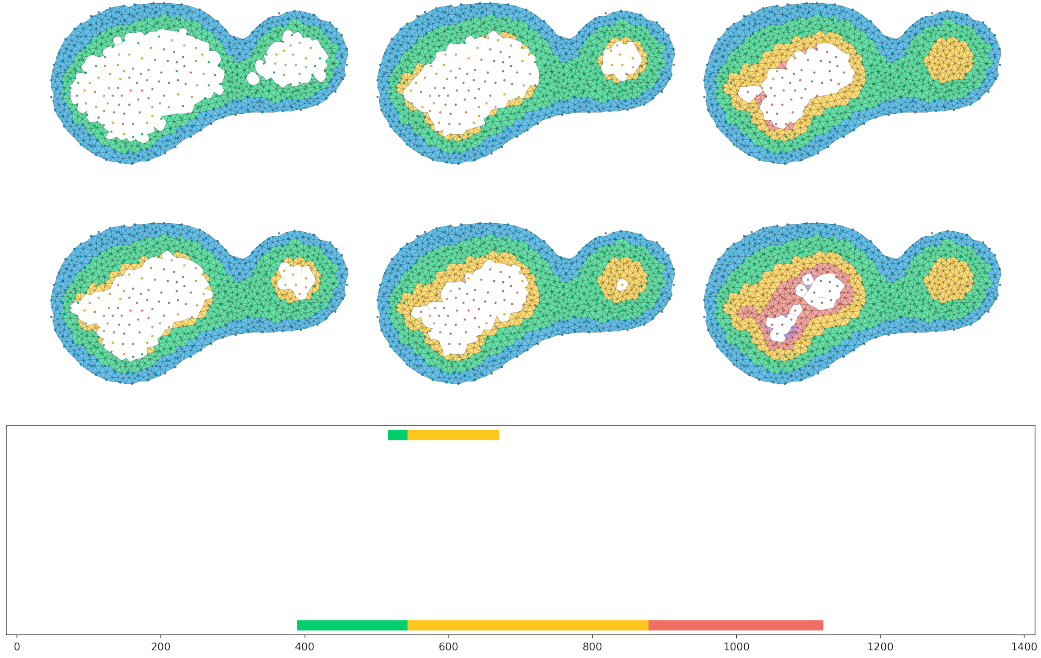
Proof. The sub-barcode relation follows from Theorem 12 and the observation that the radius $r_s \leq \varepsilon$ for all s in an ε -sample. The bottleneck bound follows from a more general theory of interleaving of image persistence modules. It depends only on the fact that the extensions u^* and ℓ^* of u and ℓ to piecewise constant functions on Voronoi cells (as in the proof of Theorem 12) satisfy $\|u^* - \check{f}_P\|_\infty \leq \varepsilon$ and $\|\ell^* - \hat{f}_P\|_\infty \leq \varepsilon$, respectively. These bounds yeild an ε -interleaving of the sublevels, and thus give an interleaving of the images.³ ◀

6.3 Finer Approximation with Barycentric Subdivision

In general, we would like to have tight approximations to the sub-barcode of Lipschitz extensions without requiring any sampling assumptions. One way to get a better bound is to manually add points to reduce the radii of the cells. Another approach would be to consider the barycentric subdivision. This makes sense because it can be understood as assigning function values to Delaunay simplices individually rather than inducing the values from the function on the vertices. In this section, we show how to define such constructions in a way that applies to a much more general class of functions.

For each sample $s \in S$, knowing the value of $f(s)$ implies an upper bound and a lower bound on f . Thus far, these bounds came from the assumption that f is Lipschitz. Similar upper and lower bounds can be defined for other classes of functions. We only assume that these bounds are *distance monotone*. That is, the upper bound \check{f}_s for a point s satisfies the condition that if $d(x, s) \leq d(y, s)$, then $\check{f}_s(x) \leq \check{f}_s(y)$. For the lower bound \hat{f}_s at s values decrease with distance, i.e., if $d(x, s) \leq d(y, s)$, then $\hat{f}_s(x) \geq \hat{f}_s(y)$. We call $\{\check{f}_s\}$ and $\{\hat{f}_s\}$ a family of *pointwise distance monotone bounds*. As before, we let $\check{f}_S(x) := \min_{s \in S} \check{f}_s(x)$ and $\hat{f}_S(x) := \max_{s \in S} \hat{f}_s(x)$.

³ A more general theorem for stability of image persistence can be found in the full version of this paper [10].



■ **Figure 6** The top row shows the Delaunay filtration of the max Lipschitz extension. The next row shows the min extension. The barcode of the image suppresses the small spurious features and captures the two main features.

Although the term barycentric subdivision implies that subdivision happen at the barycenters of the cells, we define this operation instead as a combinatorial operation on simplicial complexes. For any simplicial complex K , we define $\mathbf{bary}K$ to be the simplicial complex with a vertex for each simplex in K and a simplex for each ordered subset of simplices (the ordering is by inclusion). For any function $g : K \rightarrow \mathbb{R}$, we can extend it to a filtration G on $\mathbf{bary}K$, as $\sigma_0 \hookrightarrow \dots \hookrightarrow \sigma_k$ is in $G(t)$ if and only if $g(\sigma_i) \leq t$ for all $i \in \{0, \dots, k\}$. Proof of the following theorem may be found in the full version [10].

▶ **Theorem 14.** *Let S be a finite sample of a convex polytope X , and let $\{\check{f}_s : \text{Vor}_S(s) \rightarrow \mathbb{R} \mid s \in S\}$ and $\{\hat{f}_s : \text{Vor}_S(s) \rightarrow \mathbb{R} \mid s \in S\}$ be families of pointwise distance-monotone bounds on some unknown function $f : X \rightarrow \mathbb{R}$. Aggregate these bounds as $\check{f}_S := \bigwedge_{s \in S} \check{f}_s$ and $\hat{f}_S := \bigvee_{s \in S} \hat{f}_s$. For $\sigma \in \text{Del}_S$, let*

$$u(\sigma) := \min_{x \in \text{Vor}_S(\sigma)} \check{f}_S(x) \quad \text{and} \quad \ell(\sigma) := \min_{x \in \text{Vor}_S(\sigma)} \hat{f}_S(x).$$

Extend u and ℓ to $\mathbf{baryDel}_S$. Then, $\text{Bar}(u \geq \ell) = \text{Bar}(\check{f}_S \geq \hat{f}_S) \sqsubseteq \text{Bar}(f)$.

▶ **Example 15.** To apply this theorem to the special case of Lipschitz functions, the resulting filtration captures the usual Lipschitz upper and lower bounds restricted to the Voronoi cells. That is,

$$\check{f}_s : \text{Vor}_S(s) \rightarrow \mathbb{R} : x \mapsto f_S(s) + d(x, s) \quad \text{and} \quad \hat{f}_s : \text{Vor}_S(s) \rightarrow \mathbb{R} : x \mapsto f_S(s) - d(x, s).$$

Letting $\alpha(\sigma) := \min_{x \in \text{Vor}_S(\sigma)} d(x, \sigma)$ be the birth time of the simplex σ in the standard Delaunay filtration (also known as the α -complex filtration). Then, the corresponding filtration on $\mathbf{baryDel}_S$ is induced by

$$u(\sigma) := \max_{s \in \sigma} f_S(s) + \alpha(\sigma), \quad \text{and} \quad \ell(\sigma) := \min_{s \in \sigma} f_S(s) - \alpha(\sigma).$$

7 Categorical Constructions

The following section describes how the concepts presented above fit neatly into a category-theoretic framework. It contains some technical concepts that may unfamiliar to computational geometers.

7.1 Smoothing Persistence Modules

Given a barcode B , an ε -smoothing is a new barcode defined by eliminating bars of length at most 2ε , and replacing all other bars $[b, d]$ with a $[b + \varepsilon, d - \varepsilon]$; that is, we trim ε from the beginning and the end of each bar. The idea of a smoothed barcode naturally arises in the literature on the stability of persistence modules [7, 5, 18]. It is clear from the definition that the smoothed barcode is a sub-barcode of the original. This relationship can be expressed clearly as arising from a factorization as follows.

Let δ_- and δ_+ be order-preserving maps $\mathbb{R} \rightarrow \mathbb{R}$ with the property that, for all $t \in \mathbb{R}$, we have $\delta_-(t) \leq t \leq \delta_+(t)$. Order preserving maps between posets are functors, so the pointwise order relations on δ_- and δ_+ correspond to natural transformations

$$\varepsilon_0 : \delta_- \Rightarrow \mathbf{1}_{\mathbb{R}} \quad \text{and} \quad \varepsilon_1 : \mathbf{1}_{\mathbb{R}} \Rightarrow \delta_+.$$

So, for any persistence module $M : \mathbb{R} \rightarrow \mathbf{vec}$, composition with M yields the following factorization.

$$\begin{array}{ccc}
 M\varepsilon_0 & \xrightarrow{\quad} & M\varepsilon_1 \\
 & \searrow & \nearrow \\
 & \mathbf{1}_M &
 \end{array} \tag{4}$$

Letting $\varepsilon = \varepsilon_1 \circ \varepsilon_0$, we can view $M\varepsilon$ (or equivalently, its image) as the smoothed persistence module. The factorization induces the sub-barcode matching in the obvious way:

$$\text{Bar}(M\varepsilon) \sqsubseteq \text{Bar}(M).$$

From this perspective, an interleaving of persistence modules may be viewed as a pair of compatible factorizations rather than a pair of compatible (shifted) homomorphisms. In fact, the pair (δ_-, δ_+) corresponds to an *adjunction* with counit ε_0 and unit ε_1 (see [14]).

7.2 Sub-Barcodes are Subobjects

Within the literature, there are several different categorifications of persistence barcodes (or equivalently persistence diagrams). Here, we identify a natural way to define a category of barcodes that plays nicely with our treatment of sub-barcodes.

First, recall that unlike much of the prior work, we do not regard barcodes as multisets. Instead, we define a barcode to be a function from a set to the intervals. In computer programming terms, we have an “is a” versus “has a” distinction; in the multiset view, each bar *is an* interval. In our view, each bar *has an* interval.

A standard categorical approach to construct a category $\int_{\mathbf{C}} F$ from a (pseudo)functor $\mathbf{C} \rightarrow \mathbf{Cat}$ is the Grothendieck construction. Let $F : \mathbf{C}^{\text{op}} \rightarrow \mathbf{Cat}$ be a functor from a category \mathbf{C} to the category \mathbf{Cat} of (small) categories. For the purposes of this work, the (*covariant*) Grothendieck construction yields a category $\int_{\mathbf{C}} F$ with objects given by pairs (C, x) for each $C \in \mathbf{C}$ and $x \in F(C)$, and arrows $(f, h) : (C, x) \rightarrow (C', x')$ for each pair of arrows $f : C \rightarrow C'$ in \mathbf{C} and $h : x \rightarrow F[f](x')$ in $F(C)$.

35:14 A Theory of Sub-Barcodes

Let $[-, \mathcal{I}] : \mathbf{Set} \rightarrow \mathbf{Cat}$ be the (pseudo)functor that takes a set X and returns the category $\mathcal{I}^X = [X, \mathcal{I}]$ of functors $X \rightarrow \mathcal{I}$. Then the category **Bar** of barcodes described in Section 3 is a contravariant Grothendieck construction

$$\mathbf{Bar} = \int_{\mathbf{Set}} [-, \mathcal{I}].$$

The objects of **Bar** are pairs $(\underline{B}, B : \underline{B} \rightarrow \mathcal{I})$ – i.e. barcodes – and a morphism of **Bar** $(\underline{A}, A) \rightarrow (\underline{B}, B)$ is a function $\phi : \underline{A} \rightarrow \underline{B}$ such that for all $\alpha \in \underline{A}$, we have that $A(\alpha) \subseteq B(\phi(\alpha))$. To see this as a Grothendieck construction, it suffices to observe that the barwise inclusion ordering is a natural transformation $A \Rightarrow B\phi$ in $[\underline{A}, \mathcal{I}]$.

The condition on morphisms ensures that a bar α can only map to a bar β when $A(\alpha) \subseteq B(\beta)$. Thus, a injective morphism is exactly a sub-barcode matching. These injective morphisms are the monomorphisms in this category and thus sub-barcodes are subobjects in this category.

► **Remark 16.** The category **Bar** defined above is also known as the category of \mathcal{I} -fuzzy sets [15], and the construction of a category of fuzzy sets using the Grothendieck construction also noted by Jardine [16]. It is also possible to replace **Set** with **Mch**, the category of sets with partial injective maps (i.e., matchings), to construct barcodes as $\int_{\mathbf{Mch}} [-, \mathcal{I}]$, or more generally, to replace **Set** with the category $\mathbf{Set}_+ \supset \mathbf{Mch}$ of sets and *partial functions*. In both cases, the monomorphisms are injective sub-barcode matchings that yield sub-barcode relations.

7.3 Ranks via Presheaves

The rank functor was previously constructed directly from a persistence module, or more generally, a factorization of persistence module homomorphisms. In this section, we show how the rank functor can also be constructed from a barcode. This is already well-known; the novelty here is that we show how the rank functor can be factored through the category of barcodes. This gives the more abstract proof that sub-barcodes are more discriminating than ranks.

For any category **C**, there is a functor $L : \int_{\mathbf{Set}} [-, \mathbf{C}] \rightarrow [\mathbf{C}^{\text{op}}, \mathbf{Set}]$ given for $A : \underline{A} \rightarrow \mathbf{C}$ by

$$LA(c) = \{a \in \underline{A} \mid \exists c \rightarrow A(a)\}.$$

The morphisms $L(A)(c \rightarrow c')$ are inclusions $L(A)(c') \subseteq L(A)(c)$. For $(f, \varepsilon) : A \rightarrow B$ in $\int_{\mathbf{Set}} [-, \mathbf{C}]$, it is easy to check that f gives a natural transformation $LA \Rightarrow LB$.

For the special case where $\mathbf{C} = \mathcal{I}$, this is a functor from barcodes to presheaves of intervals. The intuitive meaning is that, for a barcode B , the presheaf LB maps an interval J to the set of bars in B that contain all of J . All morphisms in the image of L are inclusions, and so, for $J \supseteq I$, we have $LB(J) \subseteq LB(I)$ and thus $|LB(J)| \leq |LB(I)|$. Here the vertical bars indicate set cardinality which is a functor \mathbf{card} from the poset of finite sets with inclusions to the poset \mathbb{N} . Thus, we have a functor $\mathbf{card} \circ L : \mathbf{Bar} \rightarrow \mathbf{Set}^{\mathcal{I}^{\text{op}}}$.

It is now an easy exercise to show that $\mathbf{rank} = \mathbf{card} \circ L \circ \mathbf{Bar}$. All morphisms in the functor category $[\mathcal{I}^{\text{op}}, \mathbb{N}] = \mathbb{N}^{\mathcal{I}^{\text{op}}}$ are monomorphisms, so clearly a sub-barcode relation (a monomorphism of barcodes) yields a subrank relation (a monomorphism of rank invariants). The other direction doesn't hold. The most natural way to define a barcode from the rank invariant does not give a functor to barcodes. The example is given in the proof of Theorem 6.

► **Remark 17.** Barr [1] showed that, when the indexing poset L is a Heyting algebra, the category of L -fuzzy sets is equivalent to a category of sheaves of monomorphisms. Following this result, many of the recent applications of fuzzy sets [21, 19] have defined fuzzy sets

directly as functors $L \rightarrow \mathbf{Set}$. Unfortunately, because the poset of intervals \mathcal{I} is not a Heyting algebra, this equivalence cannot be applied directly to the category \mathbf{Bar} . On the other hand, the category of intervals with the product or Egli-Milner ordering is a Heyting algebra. Extending Barr's equivalence to a category of barcodes and partial functions is the subject of future work.

References

- 1 Michael Barr. Fuzzy set theory and topos theory. *Canadian Mathematical Bulletin*, 29(4):501–508, 1986.
- 2 Ulrich Bauer, Michael Kerber, Fabian Roll, and Alexander Rolle. A unified view on the functorial nerve theorem and its variations. *Expositiones Mathematicae*, 41(4):125503, 2023. doi:10.1016/j.exmath.2023.04.005.
- 3 Ulrich Bauer and Michael Lesnick. Induced matchings and the algebraic stability of persistence barcodes. *Journal of Computational Geometry*, page Vol. 6 No. 2 (2015): Special issue of Selected Papers from SoCG 2014, 2015. doi:10.20382/JOCG.V6I2A9.
- 4 Ulrich Bauer and Maximilian Schmah. Efficient computation of image persistence. *arXiv preprint*, 2022. arXiv:2201.04170.
- 5 Peter Bubenik, Vin De Silva, and Jonathan Scott. Metrics for generalized persistence modules. *Foundations of Computational Mathematics*, 15(6):1501–1531, 2015. doi:10.1007/S10208-014-9229-5.
- 6 Mickaël Buchet, Frédéric Chazal, Tamal K. Dey, Fengtao Fan, Steve Y. Oudot, and Yusu Wang. Topological analysis of scalar fields with outliers. In *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 827–841. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICS.SOCG.2015.827.
- 7 Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry, SCG '09*, pages 237–246, New York, NY, USA, 2009. ACM. doi:10.1145/1542362.1542407.
- 8 Frédéric Chazal, Leonidas J. Guibas, Steve Y. Oudot, and Primoz Skraba. Scalar field analysis over point cloud data. *Discrete & Computational Geometry*, 46(4):743–775, 2011. doi:10.1007/S00454-011-9360-X.
- 9 Frédéric Chazal and Steve Yann Oudot. Towards persistence-based reconstruction in euclidean spaces. In *Proceedings of the Twenty-fourth Annual Symposium on Computational Geometry, SCG '08*, pages 232–241, New York, NY, USA, 2008. ACM. doi:10.1145/1377676.1377719.
- 10 Oliver A. Chubet, Kirk P. Gardner, and Donald R. Sheehy. A theory of sub-barcodes, 2022. doi:10.48550/arXiv.2206.10504.
- 11 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. doi:10.1007/S10208-010-9060-6.
- 12 David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Dmitriy Morozov. Persistent homology for kernels, images, and cokernels. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 2009.
- 13 Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, November 2002. doi:10.1007/s00454-002-2885-2.
- 14 Kirk Patrick Gardner. *Verified Topological Data Analysis and a Theory of Sub-Barcodes*. PhD thesis, North Carolina State University, 2022.
- 15 Joseph A Goguen. L-fuzzy sets. *Journal of mathematical analysis and applications*, 18(1):145–174, 1967.
- 16 John F. Jardine. Fuzzy sets and presheaves. *Compositionality*, 1:3, December 2019. doi:10.32408/compositionality-1-3.

35:16 A Theory of Sub-Barcodes

- 17 F William Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del seminario matematico e fisico di Milano*, 43:135–166, 1973.
- 18 Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015. doi:10.1007/S10208-015-9255-Y.
- 19 Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint*, 2018. arXiv:1802.03426.
- 20 Dmitry Morozov. *Homological Illusions of Persistence and Stability*. PhD thesis, Duke University, 2008.
- 21 David I Spivak. Metric realization of fuzzy simplicial sets. Self-published Notes.