Computational Topology for Data Analysis: Notes from Book by

Tamal Krishna Dey
Department of Computer Science
Purdue University
West Lafayette, Indiana, USA 46907

Yusu Wang
Halıcıoğlu Data Science Institute
University of California, San Diego
La Jolla, California, USA 92093

Topic 15: Multiparameter Persistence and Distances

We have seen that persistence modules are important objects of study in topological data analysis in that they serve as an intermediate between the raw input data and the output summarization with persistence diagrams. For 1-parameter case, the distances between modules can be computed from bottleneck distances between the corresponding persistence diagrams. For multiparameter persistence modules, we already that the indecomposables which are analogues to bars in 1-parameter case are more complicated. So, defining distances between persistence mod-

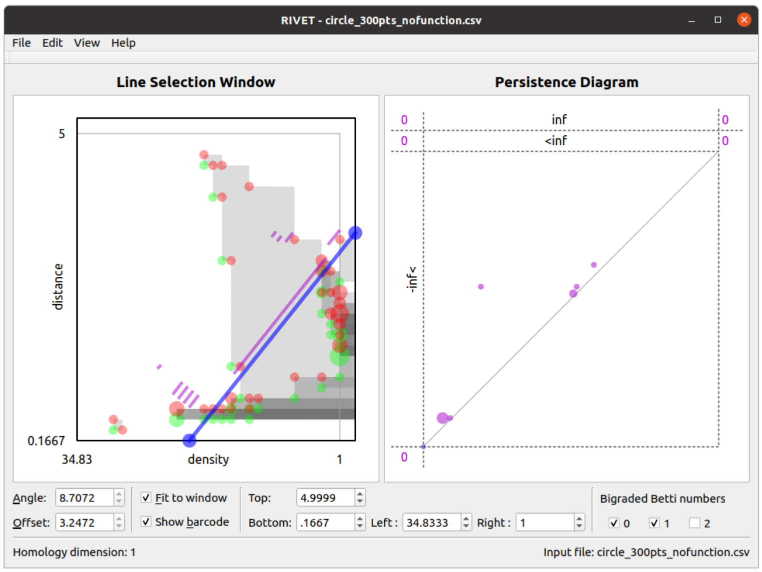


Figure 15.1: A 2-parameter module is sliced by lines that provide matching distance between two modules as we explain in Section 15.3. Figure is courtesy to RIVET software due to [22].

ules in terms of indecomposables become also more complicated. However, we need distance or distance-like notion between persistence modules to compare the input data inducing them. Figure 15.1 shows an output of RIVET software [22] that implemented the so-called matching distance between 2-parameter persistence modules. In this chapter, we describe some of these distances proposed in the literature and algorithms for computing them efficiently (polynomial time).

The interleaving distance d_I between 1-parameter persistence modules as defined earlier provides a useful means to compare them. Fortunately, for 1-parameter persistence modules, they can be computed exactly by computing the bottleneck distance d_b between their persistence diagrams thanks to the isometry theorem [21] (see also [1, 12]). We have seen a polynomial time algorithm $O(n^{1.5}\log n)$ for computing bottleneck distance. The status however is not so well settled for multiparameter persistence modules.

One of the difficulties facing the definition and computation of distances among multiparameter persistence modules is the fact that their indecomposables do not have a finite characterization as indicated previously. Even for finitely generated modules, this is true though a unique decomposition is guaranteed by Krull-Schmidt Theorem [16]. Despite this difficulty, one can define an interleaving distance d_I for multiparameter persistence modules which can be viewed as an extension of the interleaving distance defined for 1-parameter persistence modules. Shown by Lesnick [21], this distance is the most fundamental one because it is the most discriminative distance among persistence modules that is also stable with respect to functions or simplicial filtrations that give rise to the modules. Unfortunately, it turns out that computing d_I for n-parameter persistence modules and even approximating it within a factor less than 3 is NP-hard for $n \geq 2$. For a special case of modules called *interval modules*, d_I can be computed in polynomial time. In Section 15.2, we introduce the interleaving distance for multiparameter persistence modules. We follow it with a polynomial time algorithm [15] in Section 15.4.3 which computes d_I for 2-parameter interval modules.

To circumvent the problem of computing interleaving distances, several other distances have been proposed in the literature that is computable in polynomial time and bounds the interleaving distance either from above or below, but not both in the general case. Given the NP-hardness of approximating interleaving distance, there cannot exist any polynomial time computable distance that bounds d_I both from above and below within a constant factor of 3 unless $P = NP$. The *matching distance* d_m as defined in Section 15.3 bounds d_I from below, that is, $\mathsf{d}_m \leq \mathsf{d}_I$, and it can be computed in polynomial time.

Finally, in Section 15.4, we extend the definition of the bottleneck distance to multiparameter persistence modules. Extending the concept from 1-parameter case, one can define d_b as the supremum of the pairwise interleaving distances between indecomposables under an optimal matching. Then, straightforwardly, $\mathsf{d}_I \leq \mathsf{d}_b$ but the converse is not necessarily true. It is known that no lower bound in terms of d_b for d_I may exist even for a special class of 2-parameter persistence modules called interval decomposable modules [6]. However, d_b can be useful as a reasonable upper bound to d_I. Unfortunately, a polynomial time algorithm for computing d_b is not known for general persistence modules. For some persistence modules whose indecomposables have constant description such as block decomposable modules, one can compute d_b in polynomial time simply because the interleaving distance between any two modules with constant description cannot take more than $O(1)$ time.

In Section 15.4, we consider a special class of persistence modules whose indecomposables are intervals and present a polynomial time algorithm for computing d_b for them. These are modules whose indecomposables are supported by "stair-case" polyhedra. Our algorithm assumes that all indecomposables are given and computes d_b exactly for 2-parameter interval decomposable modules. Although the algorithm can be extended to persistence modules with larger number of parameters, we choose to present it only for 2-parameter case for simplicity while not losing the essential ingredients for the general case. The indecomposables required as input can be computed by the decomposition algorithm presented earlier.

15.1 Persistence modules from categorical viewpoint

In this chapter we define the persistence modules as categorical structures which are different from the graded structures used in the previous chapter. Other than introducing a different viewpoint of persistence modules, we do so because this definition becomes more amenable to defining distances. Thanks to representation theory [8, 13, 19], these two notions coincide when the modules are finitely generated in the graded module definition and are of finite type (Definition 5) in the categorical definition. Let us recall the definition in 1-parameter case. A persistence module M parameterized over $A = \mathbb{Z}$, or $\mathbb{R}$ is defined by a sequence of vector spaces M_x, $x \in A$ with linear maps $\rho_{x,y} : M_x \to M_y$ so that $\rho_{x,x}$ is identity for every $x \in A$ and for all $x, y, z \in A$ with $x \leq y \leq z$, one has $\rho_{x,z} = \rho_{y,z} \circ \rho_{x,y}$. These conditions can be formulated using category theory.

Definition 1 (Category)**.** A category $\mathcal{C}$ is a set of objects Obj $\mathcal{C}$ with a set of morphisms $\hom(x, y)$ for every pair of elements $x, y \in \text{Obj}\,\mathcal{C}$ where

1. for every $x \in \text{Obj}\,\mathcal{C}$, there is a special identity morphism $\mathbb{1}_x \in \hom(x, x)$;
2. if $f \in \hom(x, y)$ and $g \in \hom(y, z)$, then $g \circ f \in \hom(x, z)$;
3. for homomorphisms f, g, h, the compositions wherever defined are associative, that is, $(f \circ g) \circ h = f \circ (g \circ h)$;
4. $\mathbb{1}_x \circ f_{x,y} = f_{x,y}$ and $f_{x,y} \circ \mathbb{1}_y = f_{x,y}$ for every pair $x, y \in \text{Obj}\,\mathcal{C}$.

All sets form a category **Set** with functions between them playing the role of morphisms. Topological spaces form a category **Top** with continuous maps between them being the morphisms. Vector spaces form the category **Vec** with linear maps between them being the morphisms. A poset $\mathbb{P}$ form a category with every pair $x, y \in \mathbb{P}$ admitting at most one morphism; $\hom(x, y)$ has one element if $x \leq y$ and empty otherwise. Such a category is called a *thin* category in the literature for which composition rules take trivial form.

Definition 2 (Functor)**.** A functor between two categories $\mathcal{C}$ and $\mathcal{D}$ is an assignment $F : \mathcal{C} \to \mathcal{D}$ satisfying the following conditions:

1. for every $x \in \text{Obj}\,\mathcal{C}$, $F(x) \in \text{Obj}\,\mathcal{D}$;
2. for every morphism $f \in \hom(x, y)$, $F(f) \in \hom(F(x), F(y))$;
3. F respects composition, that is, $F(f \circ g) = F(f) \circ F(g)$;
4. F preserves identity morphisms, that is, $F(\mathbb{1}_x) = \mathbb{1}_{F(x)}$ for every $x \in \text{Obj}\,\mathcal{C}$.

One can observe that the 1-parameter persistence module is a functor from the category of totally ordered set of $\mathbb{Z}$ (or $\mathbb{R}$) to the category of **Vec**. Homology groups with a field coefficient for topological spaces provide a functor from category **Top** to the category of vectors spaces **Vec**. We can define maps between functors themselves.

Definition 3 (Natural transformation)**.** Given two functors $F, G : \mathcal{C} \to \mathcal{D}$, a natural transformation η from F to G, denoted as $\eta : F \implies G$, is a family of morphisms $\{\eta_x : F(x) \to G(x)\}$ for every $x \in \text{Obj}\,\mathcal{C}$ so that the following diagram commutes:

$$\begin{tikzcd} F(x) \arrow[r, "F(\rho)"] \arrow[d, "\eta_x"] & F(y) \arrow[d, "\eta_y"] \\ G(x) \arrow[r, "G(\rho)"] & G(y) \end{tikzcd}$$

Let $\mathbf{k}$ be a field, **Vec** be the category of vector spaces over $\mathbf{k}$, and **vec** be the subcategory of finite dimensional vector spaces. As usual, for simplicity, we assume $\mathbf{k} = \mathbb{Z}_2$.

Definition 4 (Persistence module)**.** Let $\mathbb{P}$ be a poset category. A $\mathbb{P}$-*indexed persistence module* is a functor $M : \mathbb{P} \to \mathbf{Vec}$. If M takes values in **vec**, we say M is *pointwise finite dimensional* (p.f.d.). The $\mathbb{P}$-indexed persistence modules themselves form another category where the natural transformations between functors constitute the morphisms.

Definition 5 (Finite type)**.** A $\mathbb{P}$-indexed persistence module M is said to have *finite type* if M is p.f.d. and all morphisms $M(x \leq y)$ are isomorphisms outside a finite subset of $\mathbb{P}$.

Here we consider the poset category to be $\mathbb{R}^n$ with the standard partial order and all modules to be of finite type. We call $\mathbb{R}^n$-indexed persistence modules as d-parameter modules in short. The reader can recognize that this is a shift from our assumption in the last chapter where we considered $\mathbb{Z}^d$-indexed modules. The category of d-parameter modules in this chapter is denoted as $\mathbb{R}^n$-**mod**. For a d-parameter module $M \in \mathbb{R}^n$-**mod**, we use notations $M_x := M(x)$ and $\rho^M_{x\to y} := M(x \leq y)$.

Definition 6 (Shift)**.** For any $\delta \in \mathbb{R}$, we denote $\vec{\delta} = (\delta, \cdots, \delta) = \delta \cdot \vec{e}$, where $\vec{e} = (e_1, e_2, \ldots, e_d)$ with $\{e_i\}_{i=1}^d$ being the standard basis of $\mathbb{R}^d$. We define a shift functor $(\cdot)_{\to\delta} : \mathbb{R}^n\text{-}\mathbf{mod} \to \mathbb{R}^n\text{-}\mathbf{mod}$ where $M_{\to\delta} := (\cdot)_{\to\delta}(M)$ is given by $M_{\to\delta}(x) = M(x + \vec{\delta})$ and $M_{\to\delta}(x \leq y) = M(x + \vec{\delta} \leq y + \vec{\delta})$. In other words, $M_{\to\delta}$ is the module M shifted diagonally by $\vec{\delta}$.

15.2 Interleaving distance

The following definition of interleaving adapts the original definition designed for 1-parameter modules in [11, 12] to d-parameter modules.

Definition 7 (Interleaving)**.** For two d-parameter persistence modules M and N, and $\delta \geq 0$, a δ-*interleaving* between M and N are two families of linear maps $\{\phi_x : M_x \to N_{x+\vec{\delta}}\}_{x\in\mathbb{R}^n}$ and $\{\psi_x : N_x \to M_{x+\vec{\delta}}\}_{x\in\mathbb{R}^n}$ satisfying the following two conditions; see Figure 15.2:

- $\forall x \in \mathbb{R}^n, \rho^M_{x\to x+2\vec{\delta}} = \psi_{x+\vec{\delta}} \circ \phi_x$ and $\rho^N_{x\to x+2\vec{\delta}} = \phi_{x+\vec{\delta}} \circ \psi_x$
- $\forall x \leq y \in \mathbb{R}^n, \phi_y \circ \rho^M_{x\to y} = \rho^N_{x+\vec{\delta}\to y+\vec{\delta}} \circ \phi_x$ and $\psi_y \circ \rho^N_{x\to y} = \rho^M_{x+\vec{\delta}\to y+\vec{\delta}} \circ \psi_x$

If such a δ-interleaving exists, we say M and N are δ-*interleaved*. We call the first condition *triangular commutativity* and the second condition *rectangular commutativity*.

Definition 8 (Interleaving distance)**.** The *interleaving distance* between modules M and N is defined as $\mathsf{d}_I(M, N) = \inf_\delta\{M \text{ and } N \text{ are } \delta\text{-}interleaved\}$. We say M and N are ∞-interleaved if they are not δ-interleaved for any $\delta \in \mathbb{R}^+$, and assign $\mathsf{d}_I(M, N) = \infty$.

The following computational hardness result from [4] is stated assuming that the input modules are represented with the graded matrices as described previously. As we mentioned before, these modules coincide with the category of modules of finite type.

Theorem 1. *Given two modules M and N given by graded matrix representations, the problem of computing a real r so that $\mathsf{d}_I(M, N) \leq r < 3\mathsf{d}_I(M, N)$ is NP-hard.*

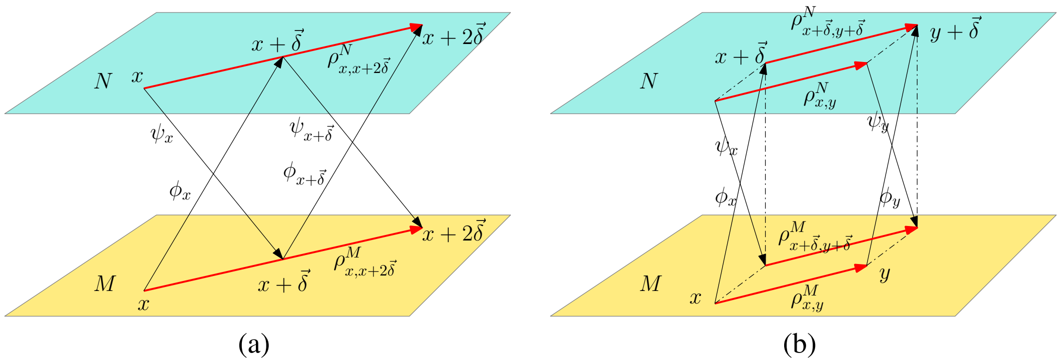


Figure 15.2: (a) Triangular commutativity, (b) Rectangular commutativity.

15.3 Matching distance

The matching distance between two persistence modules M and N draws upon the idea of taking the restrictions of M and N over lines with positive slopes and then determining the supremum of weighted interleaving distances on these restrictions. It can be defined for d-parameter modules. We are going to describe a polynomial time algorithm for computing it for 2-parameter modules, so for simplicity we define the matching distance for 2-parameter modules. Let $\ell : sx + t$ denote any line in $\mathbb{R}^2$ with $s > 0$ and let Λ denote the space of all such lines. Define a parameterization $\lambda : \mathbb{R} \to \ell$ of ℓ by taking $\lambda(x) = \frac{1}{1+s^2}(x, sx+t)$. For a line $\ell \in \Lambda$, let $M|_\ell$ denote the restriction of M on ℓ where $M|_\ell(x) = M(\lambda(x))$ with linear maps induced from M.This is a 1-parameter persistence module. We define a weight $w(\ell)$ as

$$w(\ell) = \begin{cases} \frac{1}{\sqrt{1+s^2}} & \text{for } s \geq 1 \\ \frac{1}{\sqrt{1+\frac{1}{s^2}}} & \text{for } 0 < s < 1 \end{cases}$$

Definition 9. The matching distance $\mathsf{d}_m(M, N)$ between two persistence modules is defined as

$$\mathsf{d}_m(M, N) = \sup_{\ell \in \Lambda} \{w(\ell) \cdot \mathsf{d}_I(M|_\ell, N|_\ell)\}.$$

The weight $w(\ell)$ is introduced to make the matching distance stable with respect to the interleaving distance.

15.3.1 Computing matching distance

We define a point-line duality in $\mathbb{R}^2$: a line $\ell \subset \mathbb{R}^2$ is dual to a point $\ell^* = (s, t)$ where $\ell : y = sx - t$ and a point $p = (s, t)$ is dual to a line $p^* : y = sx - t$. Following facts can be deduced from the definition easily (Exercise 3).

Fact 1.

1. *For a point p and a line ℓ, one has $(p^*)^* = p$ and $(\ell^*)^* = \ell$.*

2. *If a point p is in a line ℓ, then point ℓ^* is in line p^*.*

3. *If a point p is above (below) a line ℓ, then point ℓ^* is above (below) the line p^*.*

Consider the open half-plane Ω of $\mathbb{R}^2$ where $\Omega = \{x, y \,|\, x > 0\}$. Let α denote the bijective map between Ω and the space Λ of lines with positive slopes where $\alpha(p) = p^*$.

The representation theory [8, 13, 19] tells us that finitely generated graded modules as defined earlier are essentially equivalent to persistence modules as defined in this chapter as long as they are of finite type (Definition 5). Then, if a persistence module M is a functor on the poset $\mathbb{P} = \mathbb{R}^2$ or $\mathbb{Z}^2$, we can talk about the grades (elements of $\mathbb{P}$) of a *generating set of M and the* relations which are combinations of generators that become zero. A mindful reader can recognize these are exactly the grades of the rows and columns of the presentation matrix for M as we mentioned earlier.

Given two 2-parameter persistence modules M and N, let $\mathrm{gr}(M)$ and $\mathrm{gr}(N)$ denote the grades of all generators and relations in M and N respectively. Consider the set of lines L dual to the points in $\mathrm{gr}(M) \cup \mathrm{gr}(N)$. These lines together create a line arrangement in Ω which is a partition of Ω into vertices, edges, and faces. The vertices are points where two lines meet, the edges are maximal connected subset of the lines excluding the vertices, and faces are maximal connected subsets of Ω excluding the vertices and edges. Let A_0 denote this initial arrangement. We refine this arrangement further later. First, we observe an invariant property of the arrangemnt for which we need the following definition.

Definition 10 (Point pair type)**.** Given two points p, q and a line ℓ, we say (p, q) has the following types with respect to ℓ : (i) *Type-1* if both p and q lie above ℓ, (ii) *Type-2* if both p and q lie below ℓ, (iii) *Type-3* if p lies above and q lies below ℓ, and (iv) *Type-4* if p lies below and q lies above ℓ.

The following proposition follows from Fact 1.

Proposition 2. *For two points $p, q \in \mathrm{gr}(M) \cup \mathrm{gr}(N)$ and a face $\tau \in A_0$, the type of (p, q) with respect to the line z^* is the same for all $z \in \tau$.*

Our goal is to refine A_0 further to another arrangement A so that for every face $\tau \in A$ the grade points p, q that realizes $\mathsf{d}_I(M|_\ell, N|_\ell)$ for every $\ell = z^*$ remains the same for all $z \in \tau$. Toward that goal, we define the push of a grade point.

Definition 11 (Push)**.** For a point $p = (p_x, p_y)$ and a line $\ell : y = sx - t$, the *push* $\mathrm{push}(p, \ell)$ is defined as

$$\mathrm{push}(p, \ell) = \begin{cases} (p_x, sp_x - t) & \text{for } p \text{ below } \ell \\ ((p_y + t)/s, p_y) & \text{for } p \text{ above } \ell \end{cases}$$

Geometrically, $\mathrm{push}(p, \ell)$ is the intersection of ℓ with the upward ray originating from p in the first case, and horizontal ray originating from p in the second case. Figure 15.3 illustrates the two cases.

For $p, q \in \mathbb{R}^2$, let

$$\delta_{p,q}(\ell) = \|\mathrm{push}(p, \ell) - \mathrm{push}(q, \ell)\|_2$$

Consider the equations

$$\begin{aligned} \delta_{p,q}(\ell) &= 0 && \text{for } p, q \in \mathrm{gr}(M) \text{ or } p, q \in \mathrm{gr}(N) \\ c_{p,q}\delta_{p,q}(\ell) &= c_{p',q'}\delta_{p',q'}(\ell) && \text{for } p, q, p', q' \in \mathrm{gr}(M) \sqcup \mathrm{gr}(N) \end{aligned}$$

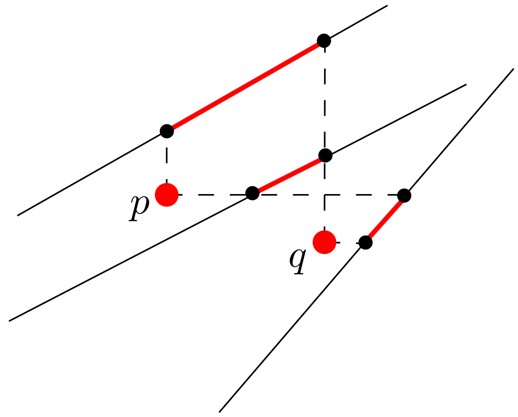


Figure 15.3: Pushes of two points to three lines. Red thick segments indicate $\delta_{p,q}$ for the corresponding lines.

where

$$c_{p,q} = \begin{cases} \frac{1}{2} & \text{if } p, q \in \mathrm{gr}(M) \text{ or } p, q \in \mathrm{gr}(N) \\ 1 & \textit{otherwise}. \end{cases}$$

The following proposition is proved in [17].

Proposition 3. *The solution set $z \in \tau$ for a face $\tau \in A_0$ so that $\delta_{p,q}(z^*)$ satisfies the above equations is either empty, the entire face τ, intersection of a line with τ, or the intersection of two lines with τ.*

Let A be the arrangement of Ω with the lines used to form A_0, the lines stated in the above proposition, and the vertical line $x = 1$.

Proposition 4. *A is formed with $O(n^4)$ lines where $n = |\mathrm{gr}(M) + \mathrm{gr}(N)|$.*

The next theorem states the main property of A which allows us to consider only finitely many (polynomially bounded) lines ℓ for computing the supremum of $\{\mathsf{d}_I(M|_\ell, N|_\ell)\}$.

Theorem 5. *For any face $\tau \in A$, there exists a pair $p, q \in \mathrm{gr}(M) \cup \mathrm{gr}(N)$ so that $c_{p,q}\delta_{p,q}(z^*) = \mathsf{d}_I(M|_{z^*}, N|_{z^*})$ for every $z \in \tau$.*

The above theorem implies that after determining the pair (p, q) for the face $\tau \in A$, we need to compute the $\sup_{z\in\tau} F(z)$ where $F(z) = \mathsf{d}_I(M|_{z^*}, N|_{z^*})$ because then considering all F over all faces in A gives the global supremum. So, now we focus on how to compute the supremum of F on a face τ.

A *region* is the closure of a face $\tau \in A$ in Ω. A region R is called *inner* if it is bounded and its closure in $\mathbb{R}^2$ does not meet the vertical line $s = 0$. See Figure 15.4. All other regions are called *outer*. An outer region has exactly two edges that are either unbounded or reaches the vertical line $s = 0$ in the limit. They are called *outer edges*. It turns out that $\sup F(z)$ is achieved either at a vertex or at the limit point of the outer edges that can be computed easily.

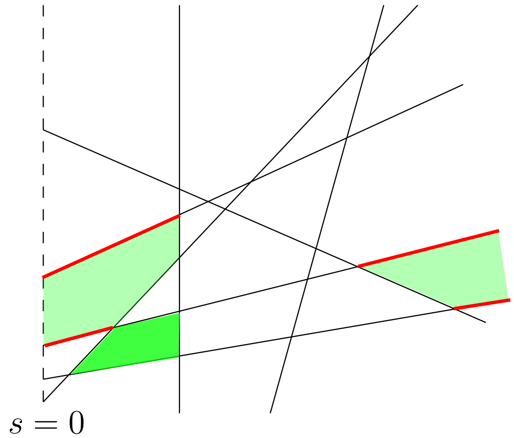


Figure 15.4: Outer regions are shaded green whose outer segments are colored red ; the region shaded yellow is inner.

Theorem 6. *The supremum* $\sup_{z \in R} F(z)$ *for a region* R *is realized either at a boundary vertex of* R *or at the limit point of an outer edge. In the latter case, let* p, q *be the pair given by Theorem 5 for* $\tau \subseteq R$*. If* e *is an outer edge and* p *lies above* z^* *for any (and all by Proposition 2)* $z \in \tau$*, then* $\sup F$ *restricted to* e *is given by:*

$$\sup F|_e = \begin{cases} |p_x - t| & \text{if line of } e \text{ intersects line } x = 0 \text{ at } t. \\ |q_x + r| & \text{if line of } e \text{ is infinite and has slope } r. \end{cases}$$

The roles of p *and* q *reverses if* p *lies below* z^* *for any* $z \in \tau$*.*

We present the entire algorithm in Algorithm 1:MatchDist. It is known that this algorithm runs in $O(n^{11})$ time where n is the total number of generators and relations for the two input modules. A more efficient algorithm approximating the matching distance is also known [18].

15.4 Bottleneck distance

Definition 12 (Matching)**.** A *matching* $\mu : A \twoheadrightarrow B$ between two multisets A and B is a partial bijection, that is, $\mu : A' \to B'$ for some $A' \subseteq A$ and $B' \subseteq B$. We say $\operatorname{im} \mu = B', \operatorname{coim} \mu = A'$.

For the next definition, we call a d-parameter module M *δ-trivial* if $\rho^M_{x \to x + \vec{\delta}} = \mathbb{0}$ for all $x \in \mathbb{R}^n$.

Definition 13 (Bottleneck distance)**.** Let $M \cong \bigoplus_{i=1}^m M_i$ and $N \cong \bigoplus_{j=1}^n N_j$ be two persistence modules, where M_i and N_j are indecomposable submodules of M and N respectively. Let $I = \{1, \cdots, m\}$ and $J = \{1, \cdots, n\}$. We say M and N are δ-matched for $\delta \geq 0$ if there exists a matching $\mu : I \twoheadrightarrow J$ so that, (i) $i \in I \setminus \operatorname{coim} \mu \implies M_i$ is 2δ-trivial, (ii) $j \in J \setminus \operatorname{im} \mu \implies N_j$ is 2δ-trivial, and (iii) $i \in \operatorname{coim} \mu \implies M_i$ *and* $N_{\mu(i)}$ *are* δ-interleaved.

The bottleneck distance is defined as

$$\mathsf{d}_b(M, N) = \inf\{\delta \mid M \text{ and } N \text{ are } \delta\text{-matched}\}.$$

Algorithm 1 MatchDist(M, N)

Input:

Two modules M and N with grades of their generators and relations

Output:

Matching distance between M and N

Compute arrangement A as described from $\mathrm{gr}(M) \cup \mathrm{gr}(N)$;
Let V be the vertex set of A;
Compute maximum $m = \max_{z \in V} F(z^*)$ over all vertices $z \in V$;
for every outer region R **do**
    Pick a point $z \in R$;
    Compute the pair $p, q \in \mathrm{gr}(M) \cup \mathrm{gr}(N)$ that realizes $\mathsf{d}_I(M|_{z^*}, N|_{z^*})$;
    if p is above z^* **then**
        if e as defined in Theorem 6 is infinite **then**
            $m := \max(m, q_x + r)$ where r is the slope of e
        else
            $m := \max(m, p_x - t)$ where e meets line $x = 0$ at t
        end if
    else
        reverse roles of p and q
    end if
end for
return m

The following fact observed in [6] is straightforward from the definition.

Fact 2. $\mathsf{d}_I \leq \mathsf{d}_b$.

15.4.1 Interval decomposable modules

We present a polynomial time algorithm for computing the bottleneck distances for a class of persistence modules called interval decomposable modules which we have seen in the previous chapter. For ease of description, we will describe the algorithm for the 2-parameter case though an extension to multiparameter case exists.

Persistence modules whose indecomposables are interval modules (Definition 15) are called *interval decomposable modules*. To account for the boundaries of free modules, we enrich the poset $\mathbb{R}^n$ by adding points at $\pm\infty$ and consider the poset $\bar{\mathbb{R}}^n = \bar{\mathbb{R}} \times \ldots \times \bar{\mathbb{R}}$ where $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ with the usual additional rule $a \pm \infty = \pm\infty$.

Definition 14 (Interval)**.** An *interval* is a subset $\emptyset \neq I \subset \bar{\mathbb{R}}^d$ that satisfies the following:

1. If $p, q \in I$ and $p \leq r \leq q$, then $r \in I$ (convexity condition);
2. If $p, q \in I$, then there exists a sequence $(p = p_0, \ldots, p_m = q) \in I$ for some $m \in \mathbb{N}$ so that for every $i \in [0, k-1]$ either $p_i \leq p_{i+1}$ or $p_i \geq p_{i+1}$ (connectivity condition). We call the sequence $(p = p_0, \ldots, p_m = q)$ a path from p to q (in I).

Let $\bar{I}$ denote the closure of an interval I in the standard topology of $\bar{\mathbb{R}}^d$. The lower and upper boundaries of I are defined as

$$\begin{aligned} L(I) &= \{x = (x_1, \cdots, x_d) \in \bar{I} \mid \forall y = (y_1, \cdots, y_d) \text{ with } y_i < x_i \ \forall i \implies y \notin I\} \\ U(I) &= \{x = (x_1, \cdots, x_d) \in \bar{I} \mid \forall y = (y_1, \cdots, y_d) \text{ with } y_i > x_i \ \forall i \implies y \notin I\}. \end{aligned}$$

Let $B(I) = L(I) \cup U(I)$. According to this definition, $\bar{\mathbb{R}}^d$ is an interval with boundary $B(\bar{\mathbb{R}}^d)$ that consists of all the points with at least one coordinate ∞. The vertex set $V(\bar{\mathbb{R}}^d)$ consists of 2^d corner points with coordinates $(\pm\infty, \cdots, \pm\infty)$.

Definition 15 (d-parameter interval module)**.** A d-parameter *interval persistence module*, or *interval module* in short, is a persistence module M that satisfies the following condition: for an interval $I_M \subseteq \bar{\mathbb{R}}^d$, called the interval of M,

$$M_x = \begin{cases} \mathbf{k} & \text{if } x \in I_M \\ \mathbf{0} & \textit{otherwise} \end{cases} \qquad \rho^M_{x \to y} = \begin{cases} \mathbb{1} & \text{if } x, y \in I_M \\ \mathbb{0} & \textit{otherwise} \end{cases}$$

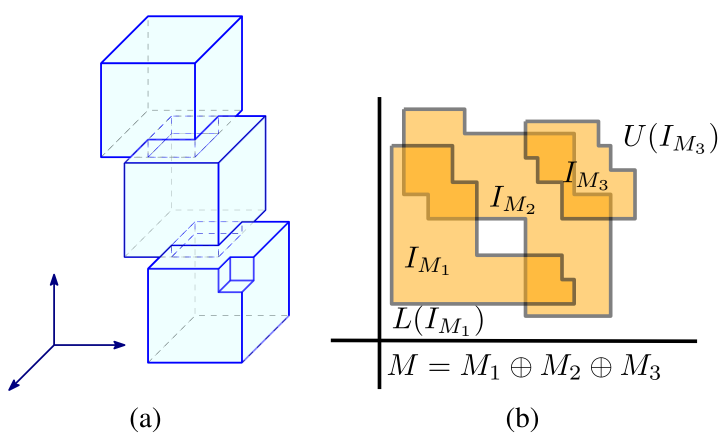
where $\mathbb{1}$ and $\mathbb{0}$ denote the identity and zero maps respectively.

It is known that an interval module is indecomposable [6].

Definition 16 (Interval decomposable module)**.** A d-parameter *interval decomposable module* is a persistence module that can be decomposed into interval modules.

Definition 17 (Rectangle)**.** A k-dimensional rectangle, $0 \leq k \leq d$, or *k-rectangle*, in $\mathbb{R}^d$, is a set $I = [a_1, b_1] \times, \cdots, \times [a_d, b_d], a_i, b_i \in \bar{\mathbb{R}}$, such that, there exists a size k index set $\Lambda \subseteq [d]$ where $\forall i \in \Lambda$, $a_i \neq b_i$, and $\forall j \in [d] \setminus \Lambda$, $a_j = b_j$.

A 0-rectangle is a vertex. A 1-rectangle is an edge. Note that a rectangle is an example of an interval.



(a) (b)

Figure 15.5: (a) Interval in $\mathbb{R}^3$, (b) Intervals in $\mathbb{R}^2$.

We say an interval $I \subseteq \bar{\mathbb{R}}^d$ is *discretely presented* if it is a finite union of d-rectangles. We also require the boundary of the interval is a $(d-1)$-manifold. A facet of I is a $(d-1)$-dimensional subset $f = \hat{f} \cap L \subseteq \bar{\mathbb{R}}^d$ where $\hat{f} = \{x_i = c\}$ is a hyperplane at some standard direction $\vec{e_i}$ in $\mathbb{R}^d$ and L is either $L(I)$ or $U(I)$. We denote the facet set as $F(I)$ and the union of all of their vertices as $V(I)$. So the boundary of I is the union of facets. And the vertices of each facet is a subset of $V(I)$. Figure 15.5(a) and (b) show intervals in $\mathbb{R}^3$ and $\mathbb{R}^2$ respectively.

For 2-parameter cases, a discretely presented interval $I \subseteq \bar{\mathbb{R}}^2$ has boundary consisting of a finite set of horizontal and vertical line segments called edges, with end points called vertices, which satisfy the following condition: (i) every vertex is incident to either a single horizontal edge or a vertical edge, (ii) no vertex appears in the interior of an edge. We denote the set of edges and vertices with $E(I)$ and $V(I)$ respectively.

We say a d-parameter interval decomposable module is finitely presented if it can be decomposed into finitely many interval modules whose intervals are discretely presented (figure on right for an example in 2-D cases). They belong to the finitely presented persistence modules as defined in the previous chapter. In the following, we focus on finitely presented interval decomposable modules.

For an interval module M, let $\overline{M}$ be the interval module defined on the closure $\overline{I_M}$. To avoid complication in this exposition, we assume that every interval module has closed intervals which is justified by the following proposition (Exercise 6).

Proposition 7. $\mathsf{d}_I(M, N) = \mathsf{d}_I(\overline{M}, \overline{N})$.

15.4.2 Bottleneck distance for 2-parameter interval decomposable modules

We present an algorithm for 2-parameter interval decomposable persistence modules though most of our definitions and claims in this section apply to general d-parameter persistence modules. They are stated and proved in the general setting wherever applicable.

Given the intervals of the indecomposables (interval modules) as input, an approach based on bipartite-graph matching is presented earlier for computing the bottleneck distance $\mathsf{d}_b(M, N)$ between two 1-parameter persistence modules M and N. This approach constructs a bipartite graph G out of the intervals of M and N and their pairwise interleaving distances including the distances to zero modules. If these distance computations take $O(C)$ time in total, then the algorithm for computing d_b takes time $O(m^{\frac{5}{2}} \log m + C)$ where M and N together have m indecomposables altogether. Observe that, the term $m^{\frac{5}{2}}$ in the complexity comes from the bipartite matching. Although this could be avoided in the 1-parameter case taking advantage of the two dimensional geometry of the persistence diagrams, we cannot do this here for determining matching among indecomposables according to Definition 13. Given indecomposables (say computed by the algorithm in previous chapter or Meataxe [23]), this approach is readily extensible to the d-parameter modules if one can compute the interleaving distance between any pair of indecomposables including the zero modules. To this end, we present an algorithm to compute the interleaving distance between two 2-parameter interval modules M_i and N_j with t_i and t_j vertices respectively on their intervals in $O((t_i + t_j)\log(t_i + t_j))$ time. This gives a total time of $O(m^{\frac{5}{2}} \log m + \sum_{i,j}(t_i + t_j)\log(t_i + t_j)) = O(m^{\frac{5}{2}} \log m + t^2 \log t)$ where t is the total number of vertices over all input intervals.

Now we focus on computing the interleaving distance between two given intervals. Given intervals I_M and I_N with t vertices, the algorithm searches a value δ so that there exists two families of linear maps from M to $N_{\to\delta}$ and from N to $M_{\to\delta}$ respectively which satisfy both triangular and square commutativity. The search is done with a binary probing: For a chosen δ from a candidate set of $O(t)$ values, the algorithm determines the direction of the search by checking two conditions called *trivializability* and *validity* on the intersections of modules M and N.

Definition 18 (Intersection module)**.** For two interval modules M and N with intervals I_M and I_N respectively let $I_Q = I_M \cap I_N$, which is a disjoint union of intervals, $\coprod I_{Q_i}$. The *intersection module* Q of M and N is $Q = \bigoplus Q_i$, where Q_i is the interval module with interval I_{Q_i}. That is,

$$Q_x = \begin{cases} \mathbf{k} & \text{if } x \in I_M \cap I_N \\ \mathbf{0} & otherwise \end{cases} \quad \text{and for } x \leq y,\ \rho^Q_{x\to y} = \begin{cases} \mathbb{1} & \text{if } x, y \in I_M \cap I_N \\ \mathbb{0} & otherwise \end{cases}$$

From the definition we can see that the support of Q, $\operatorname{supp}(Q)$, is $I_M \cap I_N$. We call each Q_i an intersection component of M and N. Write $I := I_{Q_i}$ and consider $\phi : M \to N$ to be any morphism. The following proposition says that ϕ is constant on I.

Proposition 8. *$\phi|_I \equiv a \cdot \mathbb{1}$ for some $a \in \mathbf{k}$.*

Proof.

$$\begin{tikzcd} M_{p_i} \arrow[r, "\mathbb{1}"] \arrow[d, dashed, "\phi_{p_i}"'] & M_{p_{i+1}} \arrow[d, dashed, "\phi_{p_{i+1}}"] \\ N_{p_i} \arrow[r, "\mathbb{1}"'] & N_{p_{i+1}} \end{tikzcd} \qquad \begin{tikzcd} M_{p_i} \arrow[d, dashed, "\phi_{p_i}"'] & M_{p_{i+1}} \arrow[l, "\mathbb{1}"'] \arrow[d, dashed, "\phi_{p_{i+1}}"] \\ N_{p_i} & N_{p_{i+1}} \arrow[l, "\mathbb{1}"] \end{tikzcd}$$

For any $x, y \in I$, consider a path $(x = p_0, p_1, p_2, ..., p_{2m}, p_{2m+1} = y)$ in I from x *to* y and the commutative diagrams above for $p_i \leq p_{i+1}$ (left) and $p_i \geq p_{i+1}$(right) respectively. Observe that $\phi_{p_i} = \phi_{p_{i+1}}$ in both cases due to the commutativity. Inducting on i, we get that $\phi(x) = \phi(y)$. □

Definition 19 (Valid intersection)**.** An intersection component Q_i is (M, N)-*valid* if for each $x \in I_{Q_i}$ the following two conditions hold (see Figure 15.6):

$$\text{(i) } y \leq x \text{ and } y \in I_M \implies y \in I_N, \text{ and (ii) } z \geq x \text{ and } z \in I_N \implies z \in I_M$$

Proposition 9. *Let $\{Q_i\}$ be a set of intersection components of M and N with intervals $\{I_{Q_i}\}$. Let $\{\phi_x\} : M \to N$ be the family of linear maps defined as $\phi_x = \mathbb{1}$ for all $x \in I_{Q_i}$ and $\phi_x = \mathbb{0}$ otherwise. Then ϕ is a morphism if and only if every Q_i is (M, N)-valid.*

Definition 20 (Diagonal projection and distance)**.** Let I be an interval and $x \in \bar{\mathbb{R}}^n$. Let $\Delta_x = \{x + \vec{\alpha} \mid \alpha \in \mathbb{R}\}$ denote the line called *diagonal* with slope 1 that passes through x. We define (see Figure 15.7)

$$\mathrm{dl}(x, I) = \begin{cases} \min_{y \in \Delta_x \cap I}\{\mathsf{d}_\infty(x, y) := |x - y|_\infty\} \text{ if } \Delta_x \cap I \neq \emptyset \\ +\infty \text{ otherwise.} \end{cases}$$

In case $\Delta_x \cap I \neq \emptyset$, define $\pi_I(x)$, called the *projection point* of x on I, to be the point $y \in \Delta_x \cap I$ where $\mathrm{dl}(x, I) = \mathsf{d}_\infty(x, y)$.

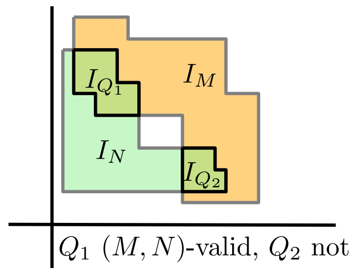


Figure 15.6: Examples of a valid intersection and a invalid intersection.

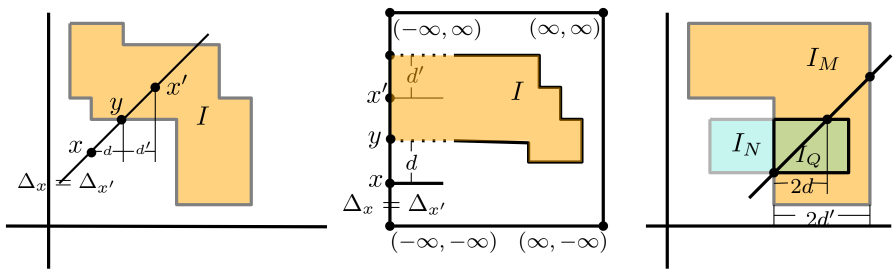


Figure 15.7: $d = \mathrm{dl}(x, I)$, $y = \pi_I(x)$, $d' = \mathrm{dl}(x', L(I))$ (left); $d = \mathrm{dl}(x, I)$ and $d' = \mathrm{dl}(x', U(I))$ are defined on the left edge of $B(\bar{\mathbb{R}}^2)$ (middle); Q is $d'_{(M,N)}$- and $d_{(N,M)}$-trivializable (right).

Note that $\forall \alpha \in \mathbb{R}$, we have $\pm\infty + \alpha = \pm\infty$. Therefore, for $x \in V(\bar{\mathbb{R}}^n)$, the line collapses to a single point. In that case, $\mathrm{dl}(x, I) \neq +\infty$ if and only if $x \in I$, which means $\pi_I(x) = x$.

Notice that upper and lower boundaries of an interval are also intervals by definition. With this understanding, following properties of dl are obvious from the above definition.

Fact 3.

(i) For any $x \in I_M$,

$$\mathrm{dl}(x, U(I_M)) = \sup_{\delta \in \bar{\mathbb{R}}}\{x + \vec{\delta} \in I_M\} \text{ and } \mathrm{dl}(x, L(I_M)) = \sup_{\delta \in \bar{\mathbb{R}}}\{x - \vec{\delta} \in I_M\}.$$

(ii) Let $L = L(I_M)$ or $U(I_M)$ and let x, x' be two points such that $\pi_L(x), \pi_L(x')$ both exist. If x and x' are on the same facet or the same diagonal line, then $|\mathrm{dl}(x, L) - \mathrm{dl}(x', L)| \leq d_\infty(x, x')$.

Set $VL(I) := V(I) \cap L(I)$, $EL(I) := E(I) \cap L(I)$, $VU(I) := V(I) \cap U(I)$, and $EU(I) := E(I) \cap U(I)$.

Proposition 10. *For an intersection component Q of M and N with interval I, the following conditions are equivalent:*

1. *Q is (M, N)-valid.*
2. *$L(I) \subseteq L(I_M)$ and $U(I) \subseteq U(I_N)$.*
3. *$VL(I) \subseteq L(I_M)$ and $VU(I) \subseteq U(I_N)$.*

Definition 21 (Trivializable intersection)**.** Let Q be a connected component of the intersection of two modules M and N. For each point $x \in I_Q$, define

$$\mathsf{d}_{triv}^{(M,N)}(x) = \max\{\mathrm{dl}(x, U(I_M))/2, \mathrm{dl}(x, L(I_N))/2)\}.$$

For $\delta \geq 0$, we say a point x is $\delta_{(M,N)}$*-trivializable* if $\mathsf{d}_{triv}^{(M,N)}(x) < \delta$. We say an intersection component Q is $\delta_{(M,N)}$*-trivializable* if each point in I_Q is $\delta_{(M,N)}$*-trivializable* (Figure 15.7). We also denote $\mathsf{d}_{triv}^{(M,N)}(I_Q) := \sup_{x \in I_Q} \{\mathsf{d}_{triv}^{(M,N)}(x)\}$.

The following proposition discretizes the search for trivializability.

Proposition 11. *An intersection component Q is $\delta_{(M,N)}$-trivializable if and only if every vertex of Q is $\delta_{(M,N)}$-trivializable.*

Recall that for two modules to be δ-interleaved, we need two families of linear maps satisfying both triangular commutativity and square commutativity. For a given δ, Theorem 14 below provides criteria which ensure that such linear maps exist. In the algorithm, we then will make sure that these criteria are verified.

Given an interval module M and the diagonal line Δ_x for any $x \in \bar{\mathbb{R}}^d$, there is a 1-parameter persistence module $M|_{\Delta_x}$ which is the functor restricted on the poset Δ_x as a subcategory of $\bar{\mathbb{R}}^d$. We call it a *1-dimensional slice* of M along Δ_x. Define

$$\delta^* = \inf_{\delta \in \bar{\mathbb{R}}} \{\delta : \forall x \in \bar{\mathbb{R}}^d, M|_{\Delta_x} \text{ and } N|_{\Delta_x} \text{ are } \delta\text{-interleaved}\}.$$

Equivalently we have $\delta^* = \sup_{x \in \bar{\mathbb{R}}^n} \{\mathsf{d}_I(M|_{\Delta_x}, N|_{\Delta_x})\}$. We have the following Proposition and Corollary from the equivalent definitions of δ^*.

Proposition 12. *For two interval modules M, N and $\delta > \delta^* \in \mathbb{R}^+$, there exist two families of linear maps $\phi = \{\phi_x : M_x \to N_{(x+\delta)}\}$ and $\psi = \{\psi_x : N_x \to M_{(x+\delta)}\}$ such that for each $x \in \bar{\mathbb{R}}^d$, the 1-dimensional slices $M|_{\Delta_x}$ and $N|_{\Delta_x}$ are δ-interleaved by the linear maps $\phi|_{\Delta_x}$ and $\psi|_{\Delta_x}$.*

Corollary 13. $\mathsf{d}_I(M, N) \geq \delta^*$.

Theorem 14. *For two interval modules M and N, $\mathsf{d}_I(M, N) \leq \delta$ if and only if the following two conditions are satisfied:*

(i) $\delta \geq \delta^*$,

(ii) $\forall \delta' > \delta$, each intersection component of M and $N_{\to\delta'}$ is either $(M, N_{\to\delta'})$-valid or $\delta_{(M,N_{\to\delta'})}$-trivializable, and each intersection component of $M_{\to\delta'}$ and N is either $(N, M_{\to\delta'})$-valid or $\delta_{(N,M_{\to\delta'})}$-trivializable.

Proof. Note that $\mathsf{d}_I(M, N) \leq \delta$ if and only if $\forall \delta' > \delta, M, N$ is δ'-interleaved.

'only if' direction: Given M and N are δ-interleaved. The part (i) follows from Corollary 13 directly. For part (ii), by definition of interleaving, $\forall \delta' > \delta$, we have two families of linear maps $\{\phi_x\}$ and $\{\psi_x\}$ which satisfy both triangular and square commutativities. Let the morphisms between the two persistence modules constituted by these two families of linear maps be $\phi = \{\phi_x\}$ and $\psi = \{\psi_x\}$ respectively. For each intersection component Q of M and $N_{\to\delta'}$ with interval

$I := I_Q$, consider the restriction $\phi|_I$. By Proposition 8, $\phi|_I$ is constant, that is, $\phi|_I \equiv \mathbb{0}$ or $\mathbb{1}$. If $\phi|_I \equiv \mathbb{1}$, by Proposition 9, Q is $(M, N_{\to\delta'})$-*valid*. If $\phi|_I \equiv \mathbb{0}$, by the triangular commutativity of ϕ, we have that $\rho^M_{x\to x+2\vec{\delta'}} = \psi_{x+\vec{\delta'}} \circ \phi_x = \mathbb{0}$ for each point $x \in I$. That means $x + 2\vec{\delta'} \notin I_M$. By Fact 3(i), $\mathrm{dl}(x, U(I_M))/2 < \delta'$. Similarly, $\rho^N_{x-\vec{\delta'}\to x+\vec{\delta'}} = \phi_x \circ \psi_{x-\vec{\delta'}} = 0 \implies x - \vec{\delta'} \notin I_N$, which is the same as to say $x - 2\vec{\delta'} \notin I_{N_{\to\delta'}}$. By Fact 3(i), $\mathrm{dl}(x, L(I_{N_{\to\delta'}}))/2 < \delta'$. So $\forall x \in I$, we have $\mathsf{d}^{(M,N_{\to\delta'})}_{triv}(x) < \delta'$. This means Q is $\delta'_{(M,N_{\to\delta'})}$-*trivializable*. Similar statement holds for intersection components of $M_{\to\delta'}$ and N.

'if' direction: We construct two families of linear maps $\{\phi_x\}, \{\psi_x\}$ as follows: On the interval $I := I_{Q_i}$ of each intersection component Q_i of M and $N_{\to\delta'}$, set $\phi|_I \equiv \mathbb{1}$ if Q_i is $(M, N_{\to\delta'})$-*valid* and $\phi|_I \equiv \mathbb{0}$ otherwise. Set $\phi_x \equiv \mathbb{0}$ for all x not in the interval of any intersection component. Similarly, construct $\{\psi_x\}$. Note that, by Proposition 9, $\phi := \{\phi_x\}$ is a morphism between M and $N_{\to\delta'}$, and $\psi := \{\psi_x\}$ is a morphism between N and $M_{\to\delta'}$. Hence, they satisfy the square commutativity. We show that they also satisfy the triangular commutativity.

We claim that $\forall x \in I_M$, $\rho^M_{x\to x+2\vec{\delta'}} = \mathbb{1} \implies x + \vec{\delta'} \in I_N$ and similar statement holds for I_N. From condition that $\delta' > \delta \geq \delta^*$ and by proposition 12, we know that there exist two families of linear maps satisfying triangular commutativity everywhere, especially on the pair of 1-parameter persistence modules $M|_{\Delta_x}$ and $N|_{\Delta_x}$. From triangular commutativity, we know that for $\forall x \in I_M$ with $\rho^M_{x\to x+2\vec{\delta'}} = \mathbb{1}$, $x+\vec{\delta'} \in I_N$ since otherwise one cannot construct a δ-interleaving between $M|_{\Delta_x}$ and $N|_{\Delta_x}$. So we get our claim.

Now for each $x \in I_M$ with $\rho^M_{x\to x+2\vec{\delta'}} = \mathbb{1}$, we have $\mathrm{dl}(x, U(I_M))/2 \geq \delta'$ by Fact 3, and $x+\vec{\delta'} \in I_N$ by our claim. This implies that $x \in I_M \cap I_{N\to\delta'}$ is a point in an interval of an intersection component Q_x of $M, N_{\to\delta'}$ which is not $\delta'_{(M,N_{\to\delta'})}$-*trivializable*. Hence, it is $(M, N_{\to\delta'})$-*valid* by the assumption. So, by our construction of ϕ on valid intersection components, $\phi_x = \mathbb{1}$. Symmetrically, we have that $x + \vec{\delta'} \in I_N \cap I_{M\to\delta'}$ is a point in an interval of an intersection component of N and $M_{\to\delta'}$ which is not $\delta'_{(N,M_{\to\delta'})}$-*trvializable* since $\mathrm{dl}(x + \vec{\delta'}, L(I_M))/2 \geq \delta'$. So by our construction of ψ on valid intersection components, $\psi_{x+\vec{\delta'}} = \mathbb{1}$. Then, we have $\rho^M_{x\to x+2\vec{\delta'}} = \psi_{x+\vec{\delta'}} \circ \phi_x$ for every nonzero linear map $\rho^M_{x\to x+2\vec{\delta'}}$. The statement also holds for any nonzero linear map $\rho^N_{x\to x+2\vec{\delta'}}$. Therefore, the triangular commutativity holds. □

Note that the above proof provides a construction of the interleaving maps for any specific δ' if it exists. Furthermore, the interleaving distance $\mathsf{d}_I(M, N)$ is the infimum of all δ' satisfying the two conditions in the theorem, which means $\mathsf{d}_I(M, N)$ is the infimum of all $\delta' \geq \delta^*$ satisfying condition 2 in Theorem 14.

15.4.3 Algorithm to compute d_I for intervals

In practice, we cannot verify all those infinitely many values $\delta' > \delta^*$. But we propose a finite candidate set of potentially possible interleaving distance values and prove later that our final target, the interleaving distance, is always contained in this finite set. Surprisingly, the size of the candidate set is only $O(n)$ with respect to the number of vertices for 2-parameter interval modules.

Based on our results, we propose a search algorithm for computing the interleaving distance $\mathsf{d}_I(M, N)$ for interval modules M and N.

Definition 22 (Candidate set)**.** For two interval modules M and N, and for each point x in $I_M \cup I_N$, let

$$
\begin{aligned}
D(x) &= \{\mathrm{dl}(x, L(I_M)), \mathrm{dl}(x, L(I_N)), \mathrm{dl}(x, U(I_M)), \mathrm{dl}(x, U(I_N))\} \text{ and} \\
S &= \{\mathsf{d} \mid \mathsf{d} \in D(x) \text{ or } 2\mathsf{d} \in D(x) \text{ for some vertex } x \in V(I_M) \cup V(I_N)\} \text{ and} \\
S_{\geq\delta} &:= \{\mathsf{d} \mid \mathsf{d} \geq \delta, \mathsf{d} \in S\}.
\end{aligned}
$$

Algorithm 2 INTERLEAVING(I_M, I_N)

Input:
I_M and I_N with t vertices in total
Output:
$\mathsf{d}_I(M, N)$

```
Compute the candidate set S and let ϵ be the half of the smallest difference between any two
numbers in S. /* O(t) time */
Compute δ*; Let δ = δ*. /* O(t) time */
Let δ* = δ_0, δ_1, ···, δ_k be numbers in S_{≥δ*} in non-decreasing order. /*O(t log t) time */
ℓ := 0; u = k;
while ℓ < u /* O(log t) probes*/ do
    i := ⌊(u+ℓ)/2⌋; δ := δ_i; δ' := δ + ε;
    Compute intersections Ω := {I_M ∩ I_{N→δ'}} ∪ {I_N ∩ I_{M→δ'}}. /* O(t) time */
    if every Q ∈ Ω is valid or trivializable according to Theorem 14 /* O(t) time*/ then
        u := i
    else
        ℓ := i
    end if
end while
Output δ
```

In Algorithm 2:INTERLEAVING, the following generic task of computing *diagonal span* is performed for several steps. Let L and U be any two chains of vertical and horizontal edges that are both x- and y-monotone. Assume that L and U have at most t vertices. Then, for a set X of $O(t)$ points in L, one can compute the intersection of Δ_x with U for every $x \in X$ in $O(t)$ total time. The idea is to first compute by a binary search a point x in X so that Δ_x intersects U if at all. Then, for other points in X, traverse from x in both directions while searching for the intersections of the diagonal line with U in lock steps.

Now we analyze the complexity of the algorithm INTERLEAVING. The candidate set, by definition, has $O(t)$ values which can be computed in $O(t)$ time by the diagonal span procedure. By Proposition 15, δ^* is in S and can be determined by computing the interleaving distances $\mathsf{d}_I(M|_{\Delta_x}, N|_{\Delta_x})$ for modules indexed by diagonal lines passing through $O(t)$ vertices of I_M and I_N. This can be done in $O(t)$ time by diagonal span procedure. Once we determine δ^*, we perform a binary search (`while` loop) with $O(\log t)$ probes for $\delta = \mathsf{d}_I(M, N)$ in the truncated set $S_{\delta\geq\delta^*}$ to satisfy the first condition of Theorem 14. Intersections between two polygons I_M and I_N bounded by x- and y-monotone chains can be computed in $O(t)$ time by a simple traversal of the boundaries. The validity and trivializability of each intersection component can be determined in time linear in the number of its vertices due to Proposition 10 and Proposition 11 respectively. Since the total number of intersection points is $O(t)$, validity check takes $O(t)$ time in total. The check for trivializabilty also takes $O(t)$ time if one uses the diagonal span procedure. So the total time complexity of the algorithm is $O(t \log t)$.

Proposition 15 below says that δ^* is determined by a vertex in I_M or I_N and $\delta^* \in S$.

Proposition 15. *(i)* $\delta^* = \max_{x \in V(I_M) \cup V(I_N)} \{\mathsf{d}_I(M|_{\Delta_x}, N|_{\Delta_x})\}$, *(ii)* $\delta^* \in S$.

The correctness of the algorithm Interleaving already follows from Theorem 14 as long as the candidate set contains the distance $\mathsf{d}_I(M, N)$. This is indeed true as shown in [15].

Theorem 16. $\mathsf{d}_I(M, N) \in S$.

Remark 1. *Our main theorem and algorithm consider the persistence modules defined on* $\mathbb{R}^2$*. For a persistence module defined on a finite or discrete poset like* $\mathbb{Z}^2$*, one can extend it to a persistence module* M *on* $\mathbb{R}^2$ *to apply our theorem and algorithm. This extension is achieved by assuming that all morphisms outside the given persistence module are isomorphisms and* $M_{x \to -\infty} = \mathbf{0}$ *if it is not given otherwise. The reader can draw the analogy between this extension and the one we had for* 1*-parameter persistence modules.*

15.5 Notes and Exercises

We already mentioned earlier that for 1-parameter persistence modules, Chazal et al. [11] showed that the bottleneck distance is bounded from above by the interleaving distance d_I; see also [5, 7, 14] for further generalizations. Lesnick [21] established the isometry theorem which showed that indeed $\mathsf{d}_I = \mathsf{d}_b$. Consequently, d_I for 1-parameter persistence modules can be computed exactly by efficient algorithms known for computing d_b.

Lesnick defined the interleaving distance for multiparameter persistence modules, and proved its stability and universality [21]. Specifically, he established that interleaving distance between persistence modules is the best discriminating distance between modules having the property of stability. It is straightforward to observe that $\mathsf{d}_I \leq \mathsf{d}_b$. For some special cases, results in the converse direction exist. Botnan and Lesnick [6] proved that, for the special class of 2-parameter persistence modules, called block decomposable modules, $\mathsf{d}_b \leq \frac{5}{2}\mathsf{d}_I$. The support of each indecomposable in such modules consists of the intersection of a bounded or unbounded axis-parallel rectangle with the upper halfplane supported by the diagonal line $x_1 = x_2$. Bjerkevic [3] improved this result to $\mathsf{d}_b \leq \mathsf{d}_I$ thereby extending the isometry theorem $\mathsf{d}_I = \mathsf{d}_b$ to 2-parameter block decomposable persistence modules.

Interestingly, a zigzag persistence module can be mapped to a block decomposable module [6]. Therefore, one can define an interleaving and a bottleneck distance between two zigzag persistence modules by the same distances on their respective block decomposable modules. Suppose that M_1 and M_2 denote the block decomposable modules corresponding to two zigzag filtration $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively. Bjerkevic's result implies that $\mathsf{d}_b(\mathrm{Dgm}_p(\mathcal{F}_1), \mathrm{Dgm}_p(\mathcal{F}_2)) \leq 2\mathsf{d}_b(M_1, M_2) = 2\mathsf{d}_I(M_1, M_2)$. The factor of 2 comes due to the difference between how distances to a null module are computed in 1-parameter and 2-parameter cases. It is important to note that the bottleneck distance d_b for persistence diagrams here takes into account the types of the bars as described for zigzag modules. This means, while matching the bars for computing this distance, only bars of similar types are matched.

A similar conclusion can also be derived for the bottleneck distance between the levelset persistence diagrams of Reeb graphs. Mapping the 0-th levelset zigzag modules $\mathbb{Z}_f, \mathbb{Z}_g$ of two Reeb graphs $(\mathbb{F}, f)$ and $(\mathbb{G}, g)$ to block decomposable modules M_f and M_g respectively, one gets that $\mathsf{d}_b(\mathrm{Dgm}_0(\mathbb{Z}_f), \mathrm{Dgm}_0(\mathbb{Z}_g)) \leq 2\mathsf{d}_b(M_f, M_g) = 2\mathsf{d}_I(M_f, M_g)$. The interleaving distance $\mathsf{d}_I(M_f, M_g)$ between block decomposable modules is bounded from above (not necessarily equal) by the interleaving distance between Reeb graphs as mentioned in earlier chapter for Reeb graphs, that is, $\mathsf{d}_I(M_f, M_g) \leq \mathsf{d}_I(\mathbb{F}, \mathbb{G})$.

Bjerkevic also extended his result to rectangle decomposable d-parameter modules (indecomposables are supported on bounded or unbounded rectangles). Specifically, he showed that $\mathsf{d}_b \leq (2d-1)\mathsf{d}_I$ for rectangle decomposable d-parameter modules and $\mathsf{d}_b \leq (d-1)\mathsf{d}_I$ for free d-parameter modules. He gave an example for exactness of this bound when $d = 2$.

Multiparameter matching distance d_m introduced in [9] provides a lower bound to interleaving distance [20]. This matching distance can be approximated within any error threshold by algorithms proposed in [2, 10]. But, it cannot provide an upper bound like d_b. The algorithm for computing d_m exactly as presented in Section 15.3 is taken from [17]. The complexity of this algorithm is rather high. To address this issue, an approximation algorithm with better time complexity has been proposed in [18] which builds on the result in [2].

For free, block, rectangle, and triangular decomposable modules, one can compute d_b by computing pairwise interleaving distances between indecomposables in constant time because they have a description of constant complexity. Due to the results mentioned earlier, d_I can be estimated within a constant or dimension-dependent factors by computing d_b for these modules. On the other hand, Botnan and Lesnick [6] observed that even for interval decomposable modules, d_b cannot approximate d_I by any constant factor approximation.

Botnan et al. [4] showed that computing interleaving distance for 2-parameter interval decomposable persistence modules as considered in this chapter is NP-hard. Worse, it cannot be approximated within a factor of 3 in polynomial time. In this context, the fact that d_b does not approximate d_I within any factor for 2-parameter interval decomposable modules [6] turns out to be a boon in disguise because otherwise a polynomial time algorithm for computing it by the algorithm as presented in Section 15.4 would not have existed. This algorithm is taken from [15] whose extension to the multiparameter persistence modules is available on arxiv.

Exercises

1. Show that d_I and d_b are pseudo-metrics on the space of finitely generated multiparameter persistence modules. Show that if the grades of generators and relations of the modules do not coincide, both become metrics.

2. Prove $\mathsf{d}_I \leq \mathsf{d}_b$ and $\mathsf{d}_m \leq \mathsf{d}_I$.

3. Prove Fact 1 for point-line duality.

4. The algorithm MatchDist computes d_m is $O(n^{11})$ time. Design an algorithm for computing d_m that runs in $o(n^{11})$ time.

5. Consider the matching distance d_m between two interval modules. Compute d_m in this case in $O(n^4)$ time.

6. Prove Proposition 7.

7. For two points $x, y \in \mathbb{R}^2$, the ℓ_∞ distance between x, y is given by $\ell_\infty(x, y) = \max\{x_1 - y_1, x_2 - y_2\}$. Given a non-negative real $\delta \geq 0$, we can define an ℓ_∞ δ-ball centered at a point $x \in \mathbb{R}^2$ as $\square_\delta(x) = \{x' \in \mathbb{R}^2 : \ell_\infty(x, x') \leq \delta\}$. We can further extend this idea to a set $I \in \mathbb{R}^2$ as $I^{+\delta} = \bigcup_{x \in I} \square_\delta(x)$, which is the union of all ℓ_∞ δ-balls centered at all points in I. For two intervals $I, J \subset \mathbb{R}^2$, the ℓ_∞ Hausdorff distance is defined as $\mathsf{d}_H(I, J) = \inf_\delta\{I \subseteq J^{+\delta}, J \subseteq I^{+\delta}\}$. Show that:

   (a) For two interval modules M and N, we have $\mathsf{d}_I(M, N) \leq \mathsf{d}_H(I_M, I_N)$.

   (b) $\mathsf{d}_I(M, N) \lneq \mathsf{d}_H(I_M, I_N)$ strictly.

   (Hint: show that $\mathsf{d}_H(I_M, I_N) \geq \delta^*$ and $\forall \delta \leq \mathsf{d}_H(I_M, I_N)$ each intersection component between $M, N_{\rightarrow\delta}$, and between $N, M_{\rightarrow\delta}$ is valid.)

Bibliography

[1] Ulrich Bauer and Michael Lesnick. Induced matchings of barcodes and the algebraic stability of persistence. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, pages 355:355–355:364, 2014.

[2] Silvia Biasotti, Andrea Cerri, Patrizio Frosini, and Daniela Giorgi. A new algorithm for computing the 2-dimensional matching distance between size functions. *Pattern Recognition Letters*, 32(14):1735–1746, 2011.

[3] Håvard Bjerkevik. Stability of higher-dimensional interval decomposable persistence modules. *arXiv preprint arXiv:1609.02086*, 2020.

[4] Håvard Bjerkevik, Magnus Botnan, and Michael Kerber. Computing the interleaving distance is np-hard. *Found. Comput. Math.*, 2019.

[5] Magnus Botnan, Justin Curry, and Elizabeth Munch. The poset interleaving distance, 2016.

[6] Magnus Botnan and Michael Lesnick. Algebraic stability of zigzag persistence modules. *Algebraic & geometric topology*, 18:3133–3204, 2018.

[7] Peter Bubenik and Jonathan Scott. Categorification of persistent homology. *Discrete & Computational Geometry*, 51(3):600–627, 2014.

[8] Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, Jul 2009.

[9] Andrea Cerri, Barbara Di Fabio, Massimo Ferri, Patrizio Frosini, and Claudia Landi. Betti numbers in multidimensional persistent homology are stable functions. *Mathematical Methods in the Applied Sciences*, 36(12):1543–1557, 2013.

[10] Andrea Cerri and Patrizio Frosini. A new approximation algorithm for the matching distance in multidimensional persistence. Technical report, February 2011.

[11] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Oudot. Proximity of persistence modules and their diagrams. In *Proc. 25th ACM Sympos. on Comput. Geom.*, pages 237–246, 2009.

[12] Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. The structure and stability of persistence modules. *arXiv preprint arXiv:1207.3674*, 2012.

[13] René Corbet and Michael Kerber. The representation theorem of persistence revisited and generalized. *Journal of Applied and Computational Topology*, 2(1):1–31, Oct 2018.

[14] Vin de Silva, Elizabeth Munch, and Amit Patel. Categorified reeb graphs. *Discrete & Computational Geometry*, 55(4):854–906, Jun 2016.

[15] Tamal K. Dey and Cheng Xin. Computing bottleneck distance for 2-d interval decomposable modules. In *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, pages 32:1–32:15, 2018.

[16] Pizzanu Kanongchaiyos and Yoshihisa Shinagawa. Articulated Reeb graphs for interactive skeleton animation. In Shuji Hashimoto, editor, *Multimedia Modeling: Modeling Multimedia Information and System*, pages 451–467. World Scientific, 2000.

[17] Michael Kerber, Michael Lesnick, and Steve Oudot. Exact computation of the matching distance on 2-parameter persistence modules. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, volume 129 of *LIPIcs*, pages 46:1–46:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[18] Michael Kerber and Arnur Nigmetov. Efficient approximation of the matching distance for 2-parameter persistence. *CoRR*, abs/1912.05826, 2019.

[19] Kevin P Knudson. A refinement of multi-dimensional persistence. *arXiv preprint arXiv:0706.2608*, 2007.

[20] Claudia Landi. The rank invariant stability via interleavings. *arXiv preprint arXiv:1412.3374*, 2014.

[21] Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015.

[22] Michael Lesnick and Matthew Wright. Interactive visualization of 2-d persistence modules. *CoRR*, abs/1512.00180, 2015.

[23] Richard A Parker. The computer calculation of modular characters (the meat-axe). *Computational group theory*, pages 267–274, 1984.