Exact Computation of the Matching Distance on 2-Parameter Persistence Modules

Michael Kerber
Graz University of Technology, Graz, Austria
kerber@tugraz.at

Michael Lesnick
University at Albany, SUNY, United States
mlesnick@albany.edu

Steve Oudot
Inria Saclay – Île-de-France, Palaiseau, France
steve.oudot@inria.fr

Abstract

The matching distance is a pseudometric on multi-parameter persistence modules, defined in terms of the weighted bottleneck distance on the restriction of the modules to affine lines. It is known that this distance is stable in a reasonable sense, and can be efficiently approximated, which makes it a promising tool for practical applications. In this work, we show that in the 2-parameter setting, the matching distance can be computed exactly in polynomial time. Our approach subdivides the space of affine lines into regions, via a line arrangement. In each region, the matching distance restricts to a simple analytic function, whose maximum is easily computed. As a byproduct, our analysis establishes that the matching distance is a rational number, if the bigrades of the input modules are rational.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Mathematics of computing → Mathematical optimization

Keywords and phrases Topological Data Analysis, Multi-Parameter Persistence, Line arrangements

Digital Object Identifier 10.4230/LIPIcs.SoCG.2019.46

Related Version A full version of this paper is available at `https://arxiv.org/abs/1812.09085`.

Funding *Michael Kerber*: Supported by Austrian Science Fund (FWF) grant number P 29984-N35.

Acknowledgements This work was initiated at the BIRS workshop “Multiparameter Persistent Homology” (18w55140) in Oaxaca, Mexico (Aug. 2018). We thank Jan Reininghaus and the other members of the discussion group on this topic for fruitful initial exchanges. We thank Matthew Wright for helpful discussions about line arrangements, slices, and the computational aspects of 2-parameter persistence.

1 Introduction

Multi-parameter persistent homology is receiving growing attention, both from the theoretical and computational points of view. Its motivation lies in the possibility of extending the success of topological data analysis to settings where the structure of data is best captured by 2-parameter rather than 1-parameter constructions. The basic algebraic objects of study in multi-parameter persistence are certain commutative diagrams of vector spaces called *persistence modules*. In the 1-parameter setting, persistence modules decompose in an essentially unique way into simple summands called *interval modules*. The decomposition is specified by a discrete invariant called a *persistence diagram*. In contrast, the algebraic

35th International Symposium on Computational Geometry (SoCG 2019).
Editors: Gill Barequet and Yusu Wang; Article No. 46; pp. 46:1–46:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

structure of a 2-parameter persistence module (henceforth, *bipersistence module*) can be far more complex. As a result, a good definition of persistence diagram is unavailable for bipersistence modules [5].

Nevertheless, it is still possible to define meaningful notions of distance between multi-parameter persistence modules. Distances on 1-parameter persistence modules play an essential role in both theory and applications. To extend such theory and applications to the multi-parameter setting, one needs to select a suitable distance on multi-parameter persistence modules. However, progress on finding well-behaved, efficiently computable distances on multi-parameter persistence modules has been slow, and this is been an impediment to progress in practical applications.

The most widely studied and applied distances in the 1-parameter setting are the *bottleneck distance* and the *Wasserstein distance* [11, 21]. Both can be efficiently computed via publicly available code [15]. In the multi-parameter setting, the distance that has received the most attention is the *interleaving distance.* On 1-parameter persistence modules, the interleaving and bottleneck distances are equal [17]. The interleaving distance is theoretically well-behaved; in particular, on modules over prime fields, it is the most discriminative distance among the ones satisfying a certain stability condition [17]. However, it was proven recently to be NP-hard to compute or even approximate to any constant factor less than three on bipersistence modules, even under the restriction that the modules decompose as direct sums of *interval modules* [4]. While the problem lies in P when further restricting the focus to the sub-class of interval modules themselves [10], in practice one often encounters modules that are not interval modules (nor even direct sums thereof).

This motivates the search for a more computable surrogate for the interleaving distance. One natural candidate is the *matching distance*, introduced by Cerri et al. [6]. It is a lower bound for the interleaving distance; this is implicit in [6] and shown explicitly in [16]. In the 2-parameter setting, the matching distance is defined as follows: Given a pair of bipersistence modules, we call an affine line ℓ in parameter space with positive slope a *slice.* Restricting the modules to ℓ yields a pair of 1-parameter persistence modules, which we call *slice modules.* These slice modules have a well-defined bottleneck distance, which we multiply by a positive weight to ensure that the matching distance will be a lower bound for the interleaving distance. The matching distance is defined as the supremum of these weighted bottleneck distances over all slices. See Section 3 for the precise definition. The definition generalizes readily to n-parameter persistence modules, for any $n \geq 1$; when $n = 1$, the matching distance is equal to the bottleneck distance.

As Cerri et al. have observed, to approximate the matching distance up to any (absolute) precision, it suffices to sample the space of slices sufficiently densely and to return the maximum weighted bottleneck distance encountered. For a constant number of scale parameters and approximation quality ϵ, a polynomial number of slices are sufficient in terms of module size and $\frac{1}{\epsilon}$, yielding a polynomial time approximation algorithm. [3]. This approach has been recently applied to the virtual ligand screening problem in computational chemistry [13]. To the best of our knowledge, there is no other previous work in which the problem of computing the matching distance has been considered.

Our contribution. We give an algorithm that computes the exact matching distance between a pair of bipersistence modules in time polynomial with respect to the size of the input. We assume that each persistence module is specified by a *presentation.* Concretely, this means that the module is specified by a matrix encoding the generators and relations, with each row and each column labeled by a point in $\mathbb{R}^2$; see Section 2.

To explain our strategy for computing the matching distance, consider the function F that assigns a slice to its weighted bottleneck distance. The matching distance is then simply the supremum of F, taken over all slices. F has a rather complicated structure, since it depends on the longest edge of a perfect matching in a bipartite graph whose edges lengths depend on both the slice and the two modules given as input. When the slice changes, the matching realizing the bottleneck distance undergoes combinatorial changes, making the function F difficult to treat analytically.

We show, however, that the space of slices can be divided into polynomially many regions so that the restriction of F to each region takes a simple closed from. Perhaps surprisingly, if we parameterize the space of slices by the right half plane $\Omega \subset \mathbb{R}^2$, the boundary between these regions can be expressed by the union of polynomially many lines in Ω, making each region convex and bounded by (possibly unbounded) line segments. (This is analogous to the observation of [18] that for a single persistence module, the locus of lines where the combinatorial structure underlying the slice module can change is described by a line arrangement.) Moreover, the restriction of F to each cell attains its supremum at a boundary vertex of the cell, or as the limit of an unbounded line segment in Ω; this follows from a straightforward case analysis. These observations together lead to a simple polynomial time algorithm to compute the matching distance.

The characterization of the matching distance underlying our algorithm also makes clear that if the row and column labels of the presentations of the input modules have rational coordinates, then the matching distance is rational as well. We are not aware of a simpler argument for this property.

Outline. We introduce the underlying topological concepts in Section 2, and introduce the matching distance in Section 3. We define the line arrangement subdividing the slice space in Section 4 and give the algorithm to maximize each cell of the arrangement in Section 5. We conclude in Section 6.

2 Persistence modules

Single-parameter modules. Let $\mathbb{K}$ be a fixed finite field throughout. A *persistence module* M over $\mathbb{R}$ is an assignment of $\mathbb{K}$-vector spaces M_x to real numbers x, and linear maps $M_{x\to y} : M_x \to M_y$ to a pair of real numbers $x \leq y$, such that $M_{x\to x}$ is the identity and $M_{y\to z} \circ M_{x\to y} = M_{x\to z}$. Equivalently, in categorical terms, a persistence module is a functor from $\mathbb{R}$ (considered as a poset category) to the category of $\mathbb{K}$-vector spaces.

A common way to arrive at a persistence module is to consider a nested sequence

$$X_1 \subseteq X_2 \subseteq \ldots \subseteq X_n$$

of simplicial complexes and to apply homology with respect to a fixed dimension and base field $\mathbb{K}$. This yields a sequence

$$H_p(X_1, \mathbb{K}) \to H_p(X_2, \mathbb{K}) \to \ldots \to H_p(X_n, \mathbb{K})$$

of vector spaces and linear maps. To obtain a persistence module over $\mathbb{R}$, we pick *grades* $s_1 < s_2 < \ldots < s_n$ and set $M_x := 0$ if $x < s_1$ and $M_x := H_p(X_i, \mathbb{K})$ with $i = \max\{j \mid s_j \leq x\}$ otherwise. For $y \geq x$ and $M_y = H_p(X_j, \mathbb{K})$, we define $M_{x\to y}$ as the map $H_p(X_i, \mathbb{K}) \to H_p(X_j, \mathbb{K})$ induced by the inclusion map $X_i \to X_j$.

Finite presentations. In this work, we restrict our attention to persistence modules that are finitely presented in the following sense. A *finite presentation* is an $k \times m$ matrix P over $\mathbb{K}$, where each row and each column is labeled by a number in $\mathbb{R}$, called the *grade*, such that if $P_{ij} \neq 0$, then $\mathrm{gr}(\mathrm{row}_i) \leq \mathrm{gr}(\mathrm{col}_j)$; here $\mathrm{gr}(-)$ denotes the grade of a row or column. We refer to the multiset of grades of all rows and columns of P simply as the *set of grades of* P, and denote this set as $\mathrm{gr}(P)$. A finite presentation gives rise to a persistence module, as we describe next. The rows of P represent the generators of the module, while the columns of P encode relations (or syzygies) on the generators. Concretely, let $e_1, \ldots, e_k$ denote the standard basis of $\mathbb{K}^k$, and for $x \in \mathbb{R}$ define the subset

$$\mathrm{Gen}_x := \{e_i \mid \mathrm{gr}(\mathrm{row}_i) \leq x\}$$

Likewise, define

$$\mathrm{Rel}_x := \{\mathrm{col}_j \mid \mathrm{gr}(\mathrm{col}_j) \leq x\}$$

Then, we define

$$M^P_x := \mathrm{span}(\mathrm{Gen}_x)/\mathrm{span}(\mathrm{Rel}_x)$$

and $M^P_{x\to y}$ simply as the map induced by the inclusion map $\mathrm{span}(\mathrm{Gen}_x) \to \mathrm{span}(\mathrm{Gen}_y)$. It can be checked easily that this indeed defines a persistence module M^P. If a persistence module N is isomorphic to M^P, we say that P is a *presentation of* N. We call a persistence module N *finitely presented* if there exists a finite presentation of N. For instance, persistence modules as above arising from a finite simplicial filtration are finitely presented. Also, the representation theorem of persistence [22, 7] states that the category of persistence modules over $\mathbb{K}$ is isomorphic to the category of graded $\mathcal{R}$-modules with an appropriately chosen ring $\mathcal{R}$. With that, a persistence module is finitely presented if and only if the corresponding $\mathcal{R}$-module is finitely presented (in the sense of a module).

We assume for concreteness that a finite presentation is encoded in terms of a sparse matrix representation [11]; the size of the presentation is understood to be the bitsize of this sparse matrix. In what follows, for finite a presentation with k generators and m relations, we will express complexity bounds in terms of $n := k + m$, the number of generators and relations. Note that an algorithm polynomial in n is also polynomial in the size of the presentation.

Persistence diagrams. A *persistence diagram* is a finite multi-set of points of the form $(b, d) \in \mathbb{R} \times (\mathbb{R} \cup \{\infty\})$ with $b < d$. A well-known structure theorem tells us that we can associate to any finitely presented persistence module M a persistence diagram $D(M)$, and that this determines M up to isomorphism [8].

Given a presentation of M, the persistence diagram can be computed by bringing the presentation matrix into echelon form. This process takes cubic time in n using Gaussian elimination [11, 22], or $O(n^\omega)$ time using fast matrix multiplication, where $\omega \leq 2.373$ [20].

▶ **Lemma 1.** *For P a finite presentation of a persistence module M and $(b, d) \in D(M)$,*

1. *b is the grade of some row of P,*
2. *if $d < \infty$, then d is the grade of some column of P.*

Proof. This follows from the correctness of the basic matrix reduction algorithm for computing persistent homology, as described in [22]. ◀

Bottleneck distance. Consider two persistence diagrams D_1 and D_2 and a bijection $\sigma : D_1' \to D_2'$ for $D_1' \subseteq D_1$ and $D_2' \subseteq D_2$. For $\delta > 0$, we define $\mathrm{cost}(\sigma) := \max(A, B)$, where

$$A = \max\{\max(|a-c|, |b-d|) \mid (a,b) \in D_1',\ \sigma(a,b) = (c,d)\},$$
$$B = \max\{(b-a)/2 \mid (a,b) \in (D_1 \setminus D_1') \cup (D_2 \setminus D_2')\},$$

and it is understood that $\infty - \infty = 0$. We define the bottleneck distance d_B by

$$d_B(D_1, D_2) = \min\{\epsilon \mid \text{there exists a matching of cost } \epsilon \text{ between } D_1 \text{ and } D_2\}.$$

For persistence modules M and N, we write $d_B(D(M), D(N))$ simply as $d_B(M, N)$.

▶ **Lemma 2.** *Let P^M and P^N be finite presentations of persistence modules M and N, respectively. $d_B(M, N)$ is equal to one of the following:*

1. *The difference of a grade of P^M and a grade of P^N,*
2. *half the difference of two grades in P^M,*
3. *half the difference of two grades in P^N.*

Proof. This follows immediately from Lemma 1 and the definition of d_B. ◀

Given two finite persistence diagrams D, D' with m points each, we can compute $d_B(D, D')$ in time to $O(m^{1.5} \log m)$ [12]; see [15] for details, including a report on practical efficiency. The number of points in a diagram is at most the number of generators of the corresponding module, and hence upper-bounded by n. Thus, the complexity of computing the bottleneck distance of two persistence modules is dominated by the computation of the persistence diagrams, and has worst-case complexity $O(n^\omega)$.

Bipersistence modules. The definitions of persistence modules and presentations extend to higher dimensions without difficulty. In the 2-parameter setting, this goes as follows: Define a partial order $\leq$ on $\mathbb{R}^2$ by $p \leq q$ if $p_x \leq q_x$ and $p_y \leq q_y$. A *bipersistence module* is an assignment of $\mathbb{K}$-vector spaces M_p to points $p \in \mathbb{R}^2$, and linear maps $M_{p\to q} : M_p \to M_q$ to pairs of points $p \leq q \in \mathbb{R}^2$, such that $M_{p\to p}$ is the identity and $M_{q\to r} \circ M_{p\to q} = M_{p\to r}$ whenever $p \leq q \leq r$. In topological data analysis, 2-dimensional persistence modules typically arise by applying homology to a bifiltered simplicial complex

A *finite presentation* of a bipersistence module $\mathbb{R}^2$ is defined in the same way as for one-parameter persistence modules, except that the labels of each row/column are elements of $\mathbb{R}^2$, and the $\leq$ relation appearing in the definition means the partial order over $\mathbb{R}^2$. From now on, we will assume that all bipersistence modules considered are finitely presented.

In topological data analysis, we do not typically have immediate access to a presentation of a bipersistence module M, but rather to a chain complex of bipersistence modules for which M is a homology module. However, it has recently been observed that from such a chain complex, a (minimal) presentation of M can be computed in cubic time [19]. The algorithm for this is practical, and has been implemented in the software package RIVET [9].

3 The matching distance

Slices. We define a *slice* as a line $\ell : y = sx + t$ where s and t are real numbers with $s > 0$. Let $\lambda : \mathbb{R} \to \ell$ be an order-preserving, isometric parameterization of the slice. Concretely, such a parameterization is given by $\lambda(x) = \frac{1}{\sqrt{1+s^2}}(x, sx + t)$. Given a bipersistence module M and slice ℓ, we define a (1-parameter) persistence module M^ℓ via $M^\ell_x := M_{\lambda(x)}$, with its linear maps induced by M. We call M^ℓ a *slice module*.

Matching distance. For a slice $\ell : y = sx + t$, we define a weight

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

$w(\ell)$ is maximized for slices with slope 1, and gets smaller when the slope goes to 0 or to ∞.

Let Λ denote the set of all slices. Writing $\Omega := (0, \infty) \times \mathbb{R}$, the map $\alpha : \Omega \to \Lambda$ that sends (s, t) to the line $y = sx + t$ is clearly a bijective parameterization of Λ. We equip Λ with the topology induced by α, using the subset topology $\Omega \subset \mathbb{R}^2$.

For two persistence modules M, N over $\mathbb{R}^2$, we define a function $F^{M,N} : \Lambda \to [0, \infty)$ by

$$F^{M,N}(\ell) := w(\ell) \cdot d_B(M^\ell, N^\ell).$$

and we define the *matching distance* between M and N as $d_{\mathrm{match}}(M, N) := \sup F^{M,N}$. As noted in the introduction, the weights $w(\ell)$ are chosen to ensure that d_{match} is a lower bound for the interleaving distance.

▶ **Lemma 3.** *Given two bipersistence modules M, N, the map $F^{M,N}$ is continuous.*

Proof. w is clearly continuous, so it suffices to show that the function $\ell \mapsto d_B(M^\ell, N^\ell)$ is continuous. Let $\mathcal{D}$ denote the metric space of all finite persistence diagrams, with metric the bottleneck distance. It follows from [16, Theorem 2] that the map sending a slice ℓ to the persistence diagram $D(M^\ell)$ is continuous with respect to the topology on $\mathcal{D}$. Thus the map sending ℓ to the pair $(D(M^\ell), D(N^\ell))$ is also continuous. Moreover, the bottleneck distance is clearly continuous as a map $\mathcal{D} \times \mathcal{D} \to [0, \infty)$, thanks to the triangle inequality. Thus, $\ell \mapsto d_B(M^\ell, N^\ell)$ is a composition of continuous functions, hence continuous. ◀

4 The arrangement

In what follows, we fix two bipersistence modules M, N (each with at most n generators and relations) and write $F := F^{M,N}$. Using the parameterization α from above, we also have a function $F \circ \alpha : \Omega \to [0, \infty)$; slightly abusing notation, we will also denote this map by F.

In this section, we construct a line arrangement in Ω in such a way that it is simple to compute $\sup F$ on each face. Recall that a *line arrangement* of Ω is the subdivision of Ω into vertices, edges, and faces induced by a finite set of distinct lines $L_1, \ldots, L_n$. The vertices of the arrangement are the intersection points of (at least) two lines, the edges are maximal connected subsets of lines not containing any vertex, and the faces are the connected components of $\Omega \setminus \bigcup_{i=1}^n L_i$. Clearly, each vertex, edge, and face of the arrangement is a convex set. The boundary of each face consists of a finite number of edges and vertices.

A first line arrangement. For $v \in \mathbb{R}^2$, let L_v denote the line $y = -v_x\, x + v_y$. Note that $L_v \cap \Omega$ is exactly the set of parameterizations of slices containing v. Now, fix presentations P^M and P^N of M and N. Let $\mathcal{A}_0$ denote the arrangement in Ω induced by the set of lines

$$\{L_v \mid v \in \mathrm{gr}(P^M) \cup \mathrm{gr}(P^N)\}.$$

In what follows, we will refine $\mathcal{A}_0$ by adding more lines into the arrangement. For this we first need to introduce some definitions.

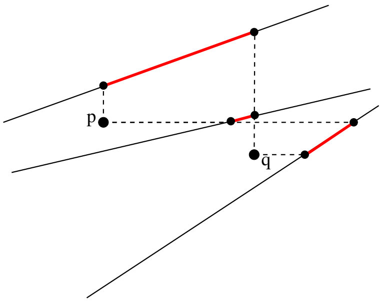

Figure 1 The pushes of two points p and q to three different slices. The length of the thick (red) line corresponds to the $\delta_{p,q}$ value of the corresponding slice.

Pushes. For a point $p = (p_x, p_y) \in \mathbb{R}^2$ and a slice $\ell : y = sx + t$, we define the *push of p onto ℓ* as

$$\operatorname{push}(p, \ell) := \begin{cases} (p_x, s \cdot p_x + t) & \text{if } p \text{ lies below } \ell \\ (\frac{p_y - t}{s}, p_y) & \text{if } p \text{ lies on or above } \ell \end{cases}$$

Geometrically, $\operatorname{push}(p, \ell)$ gives the intersection point of ℓ and a vertical upward ray emanating from p in the first case, and the intersection of ℓ with a horizontal right ray emanating from p in the second case. See Figure 1 for an illustration.

▶ Remark 4. A finite presentation of M induces a finite presentation of M^ℓ with the same underlying matrix, and each row or column grade $p \in \mathbb{R}^2$ replaced with $\lambda^{-1}(\operatorname{push}(p, \ell))$. Clearly, this presentation can be obtained in time linear in the size of the presentation of M.

For $p, q \in \mathbb{R}^2$, define $\delta_{p,q} : \Omega \to [0, \infty)$ by

$$\delta_{p,q}(s, t) := \|\operatorname{push}(p, \ell) - \operatorname{push}(q, \ell)\|_2 = |\lambda^{-1} \circ \operatorname{push}(p, \ell) - \lambda^{-1} \circ \operatorname{push}(q, \ell)|$$

with ℓ the slice defined by s and t. Again, see Figure 1 for an illustration.

We now give piecewise analytic formulae for $\delta_{p,q}$, which depend on whether the slice ℓ is below or above p and q.

(I) slice is below both p and q.

$$\delta_{p,q}(s, t) = \|(\frac{p_y - t}{s}, p_y) - (\frac{q_y - t}{s}, q_y)\|_2 = \sqrt{\left(\frac{p_y - q_y}{s}\right)^2 + (p_y - q_y)^2} = |p_y - q_y|\sqrt{1 + \frac{1}{s^2}}.$$

(II) slice is above both p and q.

$$\delta_{p,q}(s, t) = \|(p_x, sp_x + t) - (q_x, sq_x + t)\|_2 = \sqrt{(p_x - q_x)^2 + (sp_x - sq_x)^2} = |p_x - q_x|\sqrt{1 + s^2}.$$

(III) slice is between p and q. There are two subcases, which we will call (IIIa) and (IIIb): Assuming p lies above the slice (IIIa), the formula is

$$\delta_{p,q}(s, t) = \|(\frac{p_y - t}{s}, p_y) - (q_x, sq_x + t)\|_2 = \sqrt{\left(\frac{p_y - t}{s} - q_x\right)^2 + (p_y - sq_x - t)^2}$$
$$= \sqrt{\frac{1}{s^2}(p_y - t - sq_x)^2 + (p_y - sq_x - t)^2} = |p_y - t - sq_x|\sqrt{1 + \frac{1}{s^2}}.$$

If p lies below the slice (IIIb), the formula is the same, with the roles of p and q exchanged.

The push map is easily seen to be continuous with respect to the slice ℓ, so these formulae also extend to boundaries of the cases, i.e., when the slice contains p or q.

▶ **Lemma 5.** *If $p, q \in \mathrm{gr}(P^M) \cup \mathrm{gr}(P^N)$, then in each face of $\mathcal{A}_0$, exactly one of the conditions* (I), (II), (IIIa), (IIIb) *holds everywhere. Hence, $\delta_{p,q}$ can be expressed on the entire face by one of the analytic formulae above.*

Proof. Assume for a contradiction that two points x, x' in a face f of $\mathcal{A}_0$ are of two different types among (I), (II), (IIIa), (IIIb). Let $\ell = \alpha(x)$, $\ell' = \alpha(x')$ denote the corresponding slices. Then either p or q (or both) switch sides from ℓ to ℓ'. Assume that p lies above ℓ and p lies below ℓ' – all other cases are analogous. Then, on the line segment from x to x', there is some point x'' such that p lies on the slice $\alpha(x'')$. Hence, x'' lies on L_p and is therefore not in f. This contradicts the convexity of the faces. ◀

In view of Lemma 5, for $p, q \in \mathrm{gr}(P^M) \cup \mathrm{gr}(P^N)$ we may define the *type* of $\delta_{p,q}$ on a face of $\mathcal{A}_0$ to be the case (I), (II), (IIIa), or (IIIb) which holds on that face.

Refinement of the arrangement. Now we further subdivide the arrangement $\mathcal{A}_0$. For that, consider the set of equations of the form

$$\begin{aligned} \delta_{p,q}(s,t) &= 0 && \text{for } p, q \in \mathrm{gr}(P^M) \text{ or } p, q \in \mathrm{gr}(P^N), \\ c_{pq}\delta_{p,q}(s,t) &= c_{p'q'}\delta_{p',q'}(s,t) && \text{for } p, q, p', q' \in \mathrm{gr}(P^M) \sqcup \mathrm{gr}(P^N), \end{aligned} \tag{1}$$

where

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

▶ **Lemma 6.** *The solution set of each of the above equations restricted to a face f is either the empty set, the entire face, the intersection of f with a line, or intersection of f with the union of two lines.*

Proof. First we show the statement for equations of type $\delta_{p,q}(s,t) = 0$. There are 3 cases:

$\delta_{p,q}$ is of type (I). the equation becomes

$$|p_y - q_y|\sqrt{1 + \frac{1}{s^2}} = 0$$

for which either all $(s,t) \in f$ are a solution (if $p_y = q_y$), or no (s,t) is a solution.

$\delta_{p,q}$ is of type (II). the same argument holds for the equation

$$|p_x - q_x|\sqrt{1 + s^2} = 0.$$

$\delta_{p,q}$ is of type (III). Swapping p and q if necessary, we obtain the equation

$$|p_y - t - sq_x|\sqrt{1 + \frac{1}{s^2}} = 0$$

and the solution set is made of all $(s,t) \in f$ for which $p_y - t - sq_x = 0$, which is the equation of a line.

For the remaining equations, we give the proof in the special case that $c_{pq} = c_{p'q'}$; the proof in the other cases is essentially the same. For equations of the form $\delta_{p,q}(s,t) = \delta_{p',q'}(s,t)$, there are six cases to check, depending on the type of the δ-functions on the left and right sides of the equation:

Both $\delta_{p,q}(s,t)$ and $\delta_{p',q'}(s,t)$ are of type (I). the equation is

$$|p_y - q_y|\sqrt{1+\frac{1}{s^2}} = |p'_y - q'_y|\sqrt{1+\frac{1}{s^2}}$$

and the equation is satisfied if and only if $|p_y - q_y| = |p'_y - q'_y|$, independent of s and t. Hence, the solution set is either f or $\emptyset$.

Both $\delta_{p,q}(s,t)$ and $\delta_{p',q'}(s,t)$ are of type (II). the same argument as in the previous case applies, so the solution set is either f or $\emptyset$.

$\delta_{p,q}$ is of type (I) and $\delta_{p',q'}$ of type (II). we get the equation

$$|p_y - q_y|\sqrt{1+\frac{1}{s^2}} = |p'_x - q'_x|\sqrt{1+s^2}.$$

Since $1+\frac{1}{s^2} = \frac{1+s^2}{s^2}$, this simplifies to

$$|p_y - q_y| = s|p'_x - q'_x|$$

and the solution set is either all of f (if both absolute values vanish), the empty set (if only $p'_x - q'_x = 0$), or the intersection of f with the vertical line $s = \frac{|p_y - q_y|}{|p'_x - q'_x|}$ (otherwise).

Both $\delta_{p,q}$ and $\delta_{p',q'}$ are of type (III). Swapping p, q or p', q' if necessary, we get

$$|p_y - t - sq_x|\sqrt{1+\frac{1}{s^2}} = |p'_y - t - sq'_x|\sqrt{1+\frac{1}{s^2}}$$

hence, (s,t) is a solution if and only if $p_y - t - sq_x = p'_y - t - sq'_x$ or $p_y - t - sq_x = -(p'_y - t - sq'_x)$. The first equations yields again either f, $\emptyset$, or a vertical line as solution set, the second equation always defines a line. Exchanging the roles of p and q, or the roles of p' and q', or both, does not change the conclusion.

$\delta_{p,q}$ is of type (I) and $\delta_{p',q'}$ is of type (III). Swapping p' and q' if necessary, the formula is

$$|p_y - q_y|\sqrt{1+\frac{1}{s^2}} = |p_y - t - sq_x|\sqrt{1+\frac{1}{s^2}}.$$

$(s,t) \in f$ is a solution if $p_y - t - sq_x = p_y - q_y$ or $p_y - t - sq_x = q_y - p_y$, which is a line equation in both cases.

$\delta_{p,q}$ is of type (II) and $\delta_{p',q'}$ is of type (III). Swapping p' and q' if necessary, we get

$$|p_x - q_x|\sqrt{1+s^2} = |p'_y - t - sq'_x|\sqrt{1+\frac{1}{s^2}}$$

which simplifies to

$$s|p_x - q_x| = |p'_y - t - sq'_x|$$

Here $(s,t) \in f$ is a solution if and only if $s(p_x - q_x) = p'_y - t - sq'_x$ or $s(p_x - q_x) = -(p'_y - t - sq'_x)$. Again, we obtain a line in both cases. ◀

▶ **Definition 7.** *Let $\mathcal{A}$ denote the line arrangement in Ω formed by the lines in $\mathcal{A}_0$, all lines from the case analysis above, and the vertical line $s = 1$.*

▶ **Lemma 8.** *The arrangement $\mathcal{A}$ consists of $O(n^4)$ lines.*

Proof. The case analysis in the proof of Lemma 6 was performed relative to a choice of face f in $\mathcal{A}_0$. However, for a fixed choice of an equation in the set of equations (1), the lines which arise in the case analysis depend only on the types of $\delta_{p,q}$ and $\delta_{p',q'}$ on f. There are at most $4 \times 4 = 16$ possible ways of jointly choosing those types, and for a given choice, at most two lines are added to the arrangement. Hence, each of the $O(n^4)$ equations in the set of equations (1) contributes at most a constant number of lines to $\mathcal{A}$. The result now follows easily. ◀

Note that the arrangement $\mathcal{A}$ depends on the choice of presentations for M and N. The next statement says that within a face of the arrangement, the bottleneck distance is realized by the difference of the pushes of two fixed grades of the presentations.

▶ **Theorem 9.** *For any face f of $\mathcal{A}$, there is some choice of $p, q \in \mathrm{gr}(P^M) \cup \mathrm{gr}(P^N)$ such that $d_B(M^\ell, N^\ell) = c_{pq}\delta_{p,q}(\ell)$ for all $\ell \in \alpha(f)$.*

See the full version [14, App. A] for a complete proof. It can be summarized as follows: By Remark 4 and Lemma 1, for each $\ell \in f$ there is a collection T_M^ℓ of pairs

$$(b, d) \in \mathrm{gr}(P^M) \times (\mathrm{gr}(P^M) \cup \{(\infty, \infty)\})$$

such that $D(M^\ell)$ is obtained by pushing the elements of T_M^ℓ onto ℓ. Call such T_M^ℓ a *template* for $D(M^\ell)$. For ℓ and ℓ' such that the grades of P^M push onto ℓ and ℓ' in the same order, the templates for $D(M^\ell)$ and $D(M^{\ell'})$ are the same. But in fact, the grades of P^M push onto all lines $\ell \in f$ in the same order, because whenever the order changes, we need to cross one of the lines of the arrangement $\mathcal{A}$. Hence, there exists T_M^f that is a template for $D(M^\ell)$, for all $\ell \in f$. Similarly for N.

By Lemma 2, for any fixed $\ell' \in f$ we have that $d_B(M^{\ell'}, N^{\ell'}) = c_{pq}\delta_{p,q}(\ell')$, for p and q each some coordinate of a pair in $T_M^f \cup T_N^f$. The order of the values of the functions

$$\{c_{rs}\delta_{r,s} \mid r, s \in \mathrm{gr}(P^M) \cup \mathrm{gr}(P^N)\}$$

remains the same across f, because any change in the order will result again in crossing one of the lines of the arrangement $\mathcal{A}$. Thus, in fact $d_B(M^\ell, N^\ell) = c_{pq}\delta_{p,q}(\ell)$ for all $\ell \in f$.

5 Maximization

We define a *region* of $\mathcal{A}$ as the closure of a face of $\mathcal{A}$ within Ω. We can compute the matching distance by determining $\sup F(s,t)$ separately for each region in $\mathcal{A}$. We will show now that in each region, $\sup F(s,t)$ is either realized at a boundary vertex, or as the limit of an unbounded boundary edge, which can be computed easily.

We fix the following notation: A region $R \subseteq \Omega$ is an *inner region* if it is bounded as a set in $\mathbb{R}^2$ and it has a positive distance to the vertical line $s = 0$ in $\mathbb{R}^2$ (in other words, R does not reach the boundary of Ω). An inner region is a convex polygon. Regions that are not inner region are called *outer regions*. Outer regions have exactly two *outer segments* in their boundary, which are infinite or converge to a point on the vertical line $s = 0$ (except for the empty arrangement, which only consists of one outer region). See Figure 2 for an illustration.

For a fixed region R of $\mathcal{A}$, Theorem 9 ensures that there is a pair of grades (p, q) whose δ-function realizes d_B within the interior of R (a face of $\mathcal{A}$). By continuity, this implies that $\delta_{p,q}$ also realizes d_B on the entire region.

▶ **Lemma 10.** *The supremum of F within R is attained either at a vertex on the boundary of R, or as the limit of F along an outer segment. In the latter case, the limit can be expressed in simple terms based on the equation of the line segment and one of the functions $\delta_{p,q}$.*

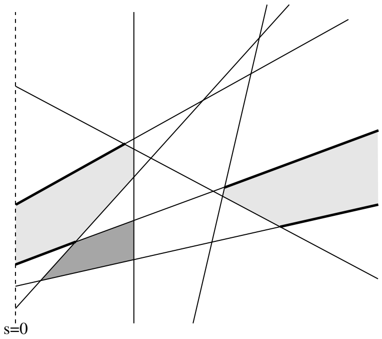

Figure 2 The lightly shaded regions show outer regions. The outer segments of these are drawn more thickly. The darkly shaded region is an example of an inner region.

Proof. We distinguish 6 cases based on the type of the δ-function and on whether $s \leq 1$ or $s \geq 1$ (note that each cell belongs to one of these cases, because the line $s = 1$ is in $\mathcal{A}$).

$\delta_{p,q}$ of type (I), $s \leq 1$. In that case,

$$F(\ell) = w(\ell)\delta_{p,q}(\ell) = \frac{1}{\sqrt{1+\frac{1}{s^2}}}|p_y - q_y|\sqrt{1+\frac{1}{s^2}} = |p_y - q_y|,$$

a constant function. Clearly, the supremum is attained everywhere, in particular at the boundary vertices of R.

$\delta_{p,q}$ of type (I), $s \geq 1$. We get

$$F(\ell) = \frac{1}{\sqrt{1+s^2}}|p_y - q_y|\sqrt{1+\frac{1}{s^2}} = \frac{1}{s}|p_y - q_y|$$

Clearly, this function becomes larger when s gets smaller. Moreover, because $s \geq 1$ within the cell, there is a leftmost vertex on the boundary, which minimizes s and therefore attains the supremum within the cell.

$\delta_{p,q}$ of type (II), $s \leq 1$. We obtain

$$F(\ell) = \frac{1}{\sqrt{1+\frac{1}{s^2}}}|p_x - q_x|\sqrt{1+s^2} = s|p_x - q_x|.$$

Similarly to the previous case, there exists a rightmost boundary vertex in the cell (because $s \leq 1$), which realizes the supremum.

$\delta_{p,q}$ of type (II), $s \geq 1$. The function simplifies to

$$F(\ell) = \frac{1}{\sqrt{1+s^2}}|p_x - q_x|\sqrt{1+s^2} = |p_x - q_x|,$$

a constant function, which attains its supremum at any boundary vertex.

$\delta_{p,q}$ of type (III), $s \leq 1$. Assuming that p lies above the slice, we get

$$F(\ell) = \frac{1}{\sqrt{1+\frac{1}{s^2}}}|p_y - t - sq_x|\sqrt{1+\frac{1}{s^2}} = |p_y - t - sq_x|.$$

If R is an inner region, we are maximizing the above function over a closed convex polygon, and the maximum is achieved at a boundary vertex, because $|p_y - t - sq_x|$ is the maximum of two linear functions in s and t.

It remains to analyze the case that R is an outer region. We argue first that R is bounded in t-direction from above and below: Since $\delta_{p,q}$ is of type (III), with p lying above ℓ, (s,t) must be below the (non-vertical) line $t = -sp_x + p_y$ in the dual space. Likewise, since q is below ℓ, (s,t) must be above $t = -sq_x + q_y$. Moreover, we have $0 < s \leq 1$. If the above lines intersect at a point r with s-value in $(0,1)$, R is contained in the triangle spanned by the two lines and the vertical line $s = 0$. Otherwise, R is contained in the trapezoid induced by these two lines and the vertical lines $s = 0$ and $s = 1$.

It follows that the two outer segments of R converge to the vertical line $s = 0$. Let $(0,t_1)$ denote the limit of the lower outer segment and $(0,t_2)$ the limit of the upper outer segment. Clearly $t_1 \leq t_2$. Let $\bar{R}$ denote the union of R with the vertical line segment from $(0,t_1)$ to $(0,t_2)$; note that $\bar{R}$ is the closure of R considered as a subset of $\mathbb{R}^2$. Observe that $|p_y - t - sq_x|$ is continuous over $\mathbb{R}^2$; therefore F can be continuously extended to $\bar{R}$. It follows that the supremum of F over $\bar{R}$ is attained at a boundary vertex, since $\bar{R}$ is a convex closed polygon. There are two cases: either the maximum is attained at a vertex of $\mathcal{A}$, or at $(0,t_1)$ or $(0,t_2)$. As we can readily see, the function values at the latter two points are $|p_y - t_1|$ and $|p_y - t_2|$, respectively. The case where p is below the slice and q is above is analyzed in the same way, with the roles of p and q swapped.

$\delta_{p,q}$ of type (III), $s \geq 1$ Assuming that p lies above the slice, we get

$$F(\ell) = \frac{1}{\sqrt{1+s^2}}|p_y - t - sq_x|\sqrt{1+\frac{1}{s^2}} = |\frac{p_y}{s} - \frac{t}{s} - q_x|$$

We first consider the case where R is an inner region, and we show that the function is maximized at a boundary vertex of R. The function $|\frac{p_y}{s} - \frac{t}{s} - q_x|$ has no local maximum over $\mathbb{R}^2$ since it is the absolute value of a linear function in t for any fixed s. Hence, the supremum over R must be attained on the boundary. We have to exclude the case that the maximum lies in the interior of an edge. For vertical edges this is obvious, because for a constant s, the function simplifies to the absolute value of a linear function in t which must be maximized at a boundary vertex. For a non-vertical line of the form $t = as + b$, plugging in this equation for t yields a function of the form

$$|\frac{p_y}{s} - \frac{as+b}{s} - q_x| = |\frac{1}{s}(p_y - b) - a - q_x|.$$

This is the absolute value of a monotone function in s and hence has no local maximum. Again, this implies that it is maximized at a boundary vertex.

Consider now the case where R is an outer region. As in the previous case, R is upper and lower bounded by two non-vertical lines, because we assume type (III). Hence, the two outer segments of R cannot be vertical; the lower outer segment has a slope r_1 and the upper outer segment has a slope r_2 with $r_1 < r_2$. We argue next that the supremum of $\frac{p_y}{s} - \frac{t}{s} - q_x$ is either attained at a boundary vertex, or equal to $|r_1 + q_x|$, or equal to $|r_2 + q_x|$. Let (s_i, t_i) denote a sequence of points in R such that $F(s_i,t_i)$ converges to the supremum. If (s_i,t_i) converges to a point in R (or has at least a convergent subsequence), it follows (similarly to the case of an inner region) that the limit point is a boundary vertex. Otherwise, we can assume (by passing to a subsequence) that the sequence s_i is unbounded. Moreover, the sequence $\frac{t_i}{s_i}$ is bounded by $[r_1, r_2]$ and therefore has a convergent subsequence with limit r'. Passing to this subsequence, we obtain that

$$\lim_{i\to\infty} F(s_i,t_i) = \lim_{i\to\infty} |\frac{p_y}{s_i} - \frac{t_i}{s_i} - q_x| = | - r' - q_x| = |r' + q_x|$$

Hence, the supremum must be of the form $|r' + q_x|$ for some $r' \in [r_1, r_2]$. On the other hand, this expression is clearly maximized for either $|r_1 + q_x|$ or $|r_2 + q_x|$, and there exist sequences attaining these values, for instance when choosing (s_i, t_i) on either of the outer segments. The case where p lies below the slice and q lies above is treated similarly, with the roles of p and q exchanged. ◀

The algorithm. We now give the algorithm to compute the matching distance:

- Compute the arrangement induced by $\mathcal{A}$ from Definition 7.
- For each vertex (s, t) in the arrangement, compute $F(s, t)$. Let m be the maximum among all the values.
- For each outer region R, pick a point (s, t) in the interior. Compute the bottleneck distance and identify a pair (p, q) of grades that realizes the bottleneck. Determine whether p and q are above or below the slice (s, t). If the region is of type (IIIa) with respect to p and q, then do the following:
  - If R is to the left of $s = 1$, compute the intersections $(0, t_1)$, $(0, t_2)$ of the outer segments of R with the vertical line $s = 0$. Set $m \leftarrow \max\{m,\ |p_y - t_1|,\ |p_y - t_2|\}$.
  - If R is to the right of $s = 1$, let r_1, r_2 denote the slopes of the outer segments of R. Set $m \leftarrow \max\{m,\ |r_1 + q_x|,\ |r_2 + q_x|\}$.

  If the region is of type (IIIb), proceed analogously, with the roles of p and q exchanged.
- Return m.

By "computing the arrangement", we mean to store the planar subdivision induced by the lines of the arrangement (e.g.[2, Ch.2]). In fact, it is possible to implement the algorithm without explicitly constructing the line arrangement $\mathcal{A}$ or even storing its entire set of vertices. This reduces the space complexity of the algorithm. See [14, App. B] for details.

▶ **Theorem 11.** *The above algorithm computes the matching distance in polynomial time.*

Proof. Correctness follows from Lemma 10: as we check all vertices of the arrangement, we cover the supremum of all inner regions. The outer regions are handled separately in the last steps of the algorithm.

Running time: recall from Lemma 8 that we have $O(n^4)$ lines in the arrangement $\mathcal{A}$. Hence, the arrangement has $O(n^8)$ vertices, $O(n^4)$ outer regions, and can be computed in $O(n^8 \log n)$ time using an extension of the Bentley-Ottman sweep-line algorithm [1]. For each vertex and each outer region, we have to compute two persistence diagrams, which can be done in $O(n^3)$ time, and a bottleneck distance whose complexity can be neglected. The remaining computations are negligible. Hence, we arrive at a $O(n^{11})$ algorithm. ◀

We remark that the algorithm can be realized entirely with rational arithmetic if all grades are rational numbers. Indeed, all lines in the arrangement have rational coefficients, and so do their intersection points. An intersection points corresponds to a slice along which we are required to compute the bottleneck distance. Recall from Section 3 that the definition of the slice ℓ module introduces a grade of $\lambda^{-1}(\text{push}(p, \ell))$ where $\lambda^{-1}(p_x, p_y) = \sqrt{1 + s^2} p_x$. Hence, these grades are not rational numbers. However, the bottleneck distance is multiplied with the weight $w(\ell)$ of the slice afterwards, and instead of doing so, one can as well scale all grades with $w(\ell)$ in advance. A simple calculation shows that this indeed turns the grades into rational values.

A simple analysis also reveals that if the input coordinates are rational and of bitsize $\leq b$, all intermediate computations in the algorithm can be performed with a bitsize of $\leq cb$, with c a (small) constant. Hence, the algorithm is strongly polynomial.

6 Discussion

We have presented the first polynomial time algorithm to exactly compute the matching distance for 2-parameter persistence modules. It is natural to ask about practicality of our approach. The large exponent of n^{11} seems discouraging at first, but we mention first that the worst-case running time of $O(n^3)$ for persistent homology is usually not appearing for real instances; indeed an almost linear behavior can be expected. Still, the large number of $O(n^4)$ lines in the arrangement constitutes a computational barrier in practice. There are several possibilities, however, to reduce this effect:

- Instead of computing the arrangement $\mathcal{A}$ globally, we could compute the intermediate arrangement $\mathcal{A}_0$ and refine each face of it separately, using only those lines that affect the δ-functions within this face.
- As a follow-up to the previous point, it might be possible to compute a smaller arrangement per face adaptively. The idea is to start at some interior point in a face of $\mathcal{A}_0$, identifying a pair (p, q) that realizes the bottleneck distance and then to determine the boundary of the region where (p, q) realizes the bottleneck distance.
- As a preprocessing step, we can minimize the input presentations, yielding presentations for the same modules with the smallest possible number of generators and relations (hence minimizing n) [19].

We pose the question of whether an implementation realizing the above ideas is competitive to an approximative, sampling-based approach for computing the matching distance.

Our algorithm needs to treat the outer edges of the arrangement $\mathcal{A}$ separately since our analysis does not rule out the possibility that the supremum is realized at the boundary of Ω. On the other hand, we are not aware of an example of two finite presentations whose matching distance is not realized by a particular slice in Ω. A proof that the supremum in the definition of the matching distance is in fact a maximum would greatly simplify our algorithm, since it would boil down to computing the intersection points of all lines and searching for the maximal F-value among them.

Finally, we have restricted attention to the case of two-parameter persistence modules. It is natural to conjecture that our algorithm extends to more parameters by constructing a hyperplane arrangement. It would be worthwhile to check this conjecture in future work.

References

1 J. Bentley and T. Ottmann. Algorithms for Reporting and Counting Geometric Intersections. *IEEE Transactions on Computers*, 28:643–647, 1979.

2 M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed. edition, 2008.

3 S. Biasotti, A. Cerri, P. Frosini, and D. Giorgi. A new algorithm for computing the 2-dimensional matching distance between size functions. *Pattern Recognition Letters*, 32(14):1735–1746, 2011.

4 H. Bjerkevik, M. Botnan, and M. Kerber. Computing the interleaving distance is NP-hard. *arXiv*, abs/1811.09165, 2018. `arXiv:1811.09165`.

5 G. Carlsson and A. Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009.

6 A. Cerri, B. Di Fabio, M. Ferri, P. Frosini, and C. Landi. Betti numbers in multidimensional persistent homology are stable functions. *Mathematical Methods in the Applied Sciences*, 36(12):1543–1557, 2013.

7 R. Corbet and M. Kerber. The representation theorem of persistence revisited and generalized. *Journal of Applied and Computational Topology*, 2(1):1–31, 2018.

8 W. Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. *Journal of Algebra and its Applications*, 14(05):1550066, 2015.

9 The RIVET Developers. RIVET: Software for visualization and analysis of 2-parameter persistent homology. `http://repo.rivet.online/`, 2014-2018.

10 T. Dey and C. Xin. Computing Bottleneck Distance for 2-D Interval Decomposable Modules. In *International Symposium on Computational Geometry (SoCG)*, pages 32:1–32:15, 2018.

11 H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction.* American Mathematical Society, Providence, RI, USA, 2010.

12 A. Efrat, A. Itai, and M. Katz. Geometry Helps in Bottleneck Matching and Related Problems. *Algorithmica*, 31(1):1–28, 2001. `doi:10.1007/s00453-001-0016-8`.

13 B. Keller, M. Lesnick, and T. L. Willke. Persistent Homology for Virtual Screening. *ChemRxiv preprint*, October 2018.

14 M. Kerber, M. Lesnick, and S. Oudot. Exact computation of the matching distance on 2-parameter persistence modules. *CoRR*, abs/1812.09085, 2018.

15 M. Kerber, D. Morozov, and A. Nigmetov. Geometry Helps to Compare Persistence Diagrams. *Journal of Experimental Algorithms*, 22:1.4:1–1.4:20, September 2017.

16 C. Landi. The Rank Invariant Stability via Interleavings. In *Research in Computational Topology*, pages 1–10. Springer International Publishing, 2018.

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

18 M. Lesnick and M. Wright. Interactive visualization of 2-D persistence modules. *arXiv:1512.00180*, 2015. `arXiv:1512.00180`.

19 M. Lesnick and M. Wright. Computing Minimal Presentations and Betti Numbers of 2-Parameter Persistent Homology. *arXiv:1902.05708*, 2019. `arXiv:1902.05708`.

20 N. Milosavljevic, D. Morozov, and P. Skraba. Zigzag persistent homology in matrix multiplication time. In *ACM Symposium on Computational Geometry (SoCG)*, pages 216–225, 2011.

21 S. Oudot. *Persistence theory: From Quiver Representation to Data Analysis*, volume 209 of *Mathematical Surveys and Monographs.* American Mathematical Society, 2015.

22 A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33(2):249–274, 2005.