

Report for the project of bimodules

Jingyi Li *

Institut Polytechnique de Paris , Palaiseau, France

jingyi.li@ip-paris.fr

September 29, 2022

1 Introduction

The main contribution of this report is an algorithm to compute the rank invariant of persistence bimodules. For pedagogical purposes, we first introduce the necessary concepts in single-parameter persistence and then generalize them to bipersistence in section 2. The algorithm description and pseudocode are given in section 3. Finally, we show some experimental results and analyze the running time quantitatively in section 4.

Keywords: topological data analysis, multiparameter persistence, rank invariant

2 Background

In this section, we first give the definition of single-parameter persistence module and how to compute its barcode decomposition with an inclusion-exclusion formula, and then we generalize them to bimodule.

2.1 Single-parameter persistence

Let U be a subset of \mathbb{R} .

Definition 2.1 (Single-parameter persistence module). *A persistence module \mathbb{V} over U is a collection of vector spaces $\{V_t\}_{t \in U}$ and linear maps $v_s^t := V(s \leq t) : V_s \rightarrow V_t$ with the property that v_s^s is the identity map and $v_t^u \circ v_s^t = v_s^u$ for all $s \leq t \leq u \in U$.*

Definition 2.2 (Filtration). *A filtration over U is a family $\mathcal{F} = (F_t)_{t \in U}$ of increasing (for inclusion) topological spaces:*

$$\forall s, t \in U, s \leq t \Rightarrow F_s \subset F_t$$

*This report was written during my project under Steve Oudot's supervision.

Then we can define a persistence module from the filtration.

Definition 2.3. Let $\mathcal{F} = (F_t)_{t \in U}$ be a filtration. We apply the homology functor H_* :

- $\forall t \in U$, we define $V_t := H_*(F_t, \mathbf{k})$
- $\forall s \leq t$, let v_s^t be the linear map induced by the canonical inclusion $F_s \xrightarrow{i} F_t$

Definition 2.4 (Interval). An interval of U is a convex subset $I \subset U$, i.e. $\forall s \leq t \leq u \in U, (s, u) \in I^2 \Rightarrow t \in I$.

Definition 2.5 (Interval module). An interval module over $I \subset U$ is a persistence module \mathbb{V} defined by:

- $V_t = \mathbf{k}$ if $t \in I$. $V_t = \{0\}$ otherwise.
- $\forall s, t \in U, v_s^t = \text{id}_{\mathbf{k}}$ if $(s, t) \in I^2$. $v_s^t = 0$ otherwise.

For example, if $I=[b,d]$, the interval module is

$$\mathbb{I}_{[b,d]} := \underbrace{\{0\} \xrightarrow{0} \dots \xrightarrow{0} \{0\}}_{t < b} \xrightarrow{0} \underbrace{\mathbf{k} \xrightarrow{\text{id}} \dots \xrightarrow{\text{id}} \mathbf{k}}_{b \leq t \leq d} \xrightarrow{0} \underbrace{\{0\} \xrightarrow{0} \dots \xrightarrow{0} \{0\}}_{d < t}$$

By convention, we set $\mathbb{I}_{\emptyset} = 0$.

Definition 2.6 (Direct sum of modules). Given $\mathbb{V} = (V_t)_t, \mathbb{W} = (W_t)_t$ two persistence modules with corresponding linear maps $(v_s^t), (w_s^t)$, we define $\mathbb{V} \oplus \mathbb{W}$ by

- $\mathbb{V} \oplus \mathbb{W} = (V_t \oplus W_t)_t$
- The corresponding linear maps are denoted by:

$$(v \oplus w)_s^t : V_s \oplus W_s \rightarrow V_t \oplus W_t$$

$$(x, y) \mapsto (v_s^t(x), w_s^t(y))$$

This definition extends naturally to any family of persistence modules $(\mathbb{V}_j)_{j \in J}$, denoted by $\mathbb{V} := \bigoplus_{j \in J} \mathbb{V}_j$.

Theorem 2.1. A single-parameter persistence module \mathbb{V} can be uniquely decomposed as a direct sum of interval modules, written as: $\mathbb{V} \simeq \bigoplus \mathbb{I}_{[b_j, d_j]}$, when all the vector spaces in \mathbb{V} are finite-dimensional. The multi-set of intervals in the decomposition is called the (persistence) barcode of \mathbb{V} , denoted as $\text{Dgm } \mathbb{V}$.

Definition 2.7 (Rank invariant). Given a persistence module \mathbb{V} over U , the rank invariant of \mathbb{V} is the collection of ranks $r(s, t) = \text{rank}(V(s \leq t))$ for all $s, t \in U$ such that $s \leq t$.

According to Theorem 2.1, we can decompose the rank invariant $\text{Rk} \mathbb{V}$ as a \mathbb{Z} -linear combination of rank invariants of interval modules:

$$\text{Rk} \mathbb{V} = \text{Rk} \left(\bigoplus_{I \in \text{Dgm} \mathbb{V}} \mathbf{k}_I \right) = \sum_{I \in \text{Dgm} \mathbb{V}} \text{Rk} \mathbf{k}_I, \quad (1)$$

where each interval $I \in \text{Dgm} \mathbb{V}$ is considered with multiplicity in the sums and where \mathbf{k}_I denotes the interval module supported on I .

The encoding in Equation (1) is unique, i.e. there is no other way to decompose $\text{Rk} \mathbb{V}$ as a \mathbb{Z} -linear combination, with positive coefficients, of rank invariants of interval modules [BOO21].

Dgm \mathbb{V} can then be represented graphically as an actual barcode like Figure 2.1, whose bars are in bijective correspondence with the elements of Dgm \mathbb{V} .

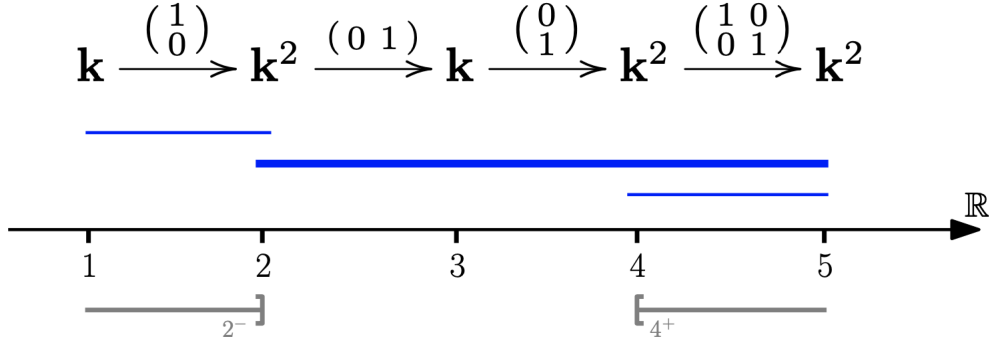


Figure 2.1: A single-parameter persistence module \mathbb{V} (top) indexed over $\{1, 2, 3, 4, 5\} \subset \mathbb{R}$, and its barcode in blue: the corresponding rank decomposition is $Rk\mathbb{V} = Rk\mathbf{k}_{[1,2]} + Rk\mathbf{k}_{[2,5]} + Rk\mathbf{k}_{[4,5]}$. For example, $Rk\mathbb{V}(2, 4) = 1$ is given by the one bar that connects the down-set 2^- to the up-set 4^+ , and $Rk\mathbb{V}(4, 5) = 2$ is given by the two bars $[2, 5]$ and $[4, 5]$.

Theorem 2.2 (Multiplicity). *When $U = \mathbb{Z}$, the multiplicity of interval $[s, t] \in Dgm\mathbb{V}$ is given by the following inclusion-exclusion formula:*

$$m(s, t) = r(s, t) - r(s - 1, t) - r(s, t + 1) + r(s - 1, t + 1). \quad (2)$$

Theorem 2.2 says that the barcode Dgm \mathbb{V} can be easily obtained from $Rk\mathbb{V}$, so \mathbb{V} is completely described up to isomorphism by the *rank invariant*, which is a desirable property for a single-parameter persistence module [CDSGO16, CSEH07].

2.2 Bipersistence

Some of the previous concepts extend naturally from single-parameter persistence to two-parameter persistence, also called bipersistence.

Let U be a subset of \mathbb{R}^2 equipped with the product order: $s \leq t$ iff $s.x \leq t.x$ and $s.y \leq t.y$.

Definition 2.8 (Bimodule). *A bimodule \mathbb{V} over U is a collection of vector spaces $\{V_t\}_{t \in U}$ and linear maps $v_s^t := V(s \leq t) : V_s \rightarrow V_t$ with the property that v_s^s is the identity map and $v_t^u \circ v_s^t = v_s^u$ for all $s \leq t \leq u \in U$, where $s \leq t$ if and only if $s.x \leq t.x$ and $s.y \leq t.y$.*

Definition 2.9 (Bifiltration). *A bifiltration over U is a family $\mathcal{F} = (F_t)_{t \in U}$ of increasing (for inclusion) topological spaces:*

$$\forall s, t \in U, s \leq t \Rightarrow F_s \subset F_t$$

Then we can define a bimodule from the bifiltration.

Definition 2.10. *Let $\mathcal{F} = (F_t)_{t \in U}$ be a bifiltration. We apply the homology functor H_* :*

- $\forall t \in U$, we define $V_t := H_*(F_t, \mathbf{k})$
- $\forall s \leq t$, let v_s^t be the linear application induced by the canonical inclusion $F_s \xrightarrow{i} F_t$

Definition 2.11 (Rank invariant). *Given a persistence module \mathbb{V} over U , the rank invariant $Rk\mathbb{V}$ is the collection of ranks $r(s, t) = \text{rank}(\mathbb{V}(s \leq t))$ for all $s, t \in U$ such that $s \leq t$.*

We also want to decompose bimodules in the same way as single-parameter persistence, so we need some analogues of intervals and interval modules in bipersistence.

Definition 2.12 (Indicator module). *For $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^2$ and a subset $\mathcal{Q} \subseteq \mathcal{X} \times \mathcal{Y}$, the indicator module of \mathcal{Q} , denoted by $k_{\mathcal{Q}}$, is defined by*

$$k_{\mathcal{Q},t} = \begin{cases} \mathbf{k}, & (t \in \mathcal{Q}) \\ 0, & (t \notin \mathcal{Q}) \end{cases} \quad (3)$$

$$k_{\mathcal{Q}}(s \leq t) = \begin{cases} Id_{\mathbf{k}}, & (s \text{ and } t \in \mathcal{Q}) \\ 0, & \text{else} \end{cases} \quad (4)$$

Definition 2.13 (Rectangle module). *For a rectangle $\mathcal{Q} \subset \mathbb{R}^2$, the rectangle module $k_{\mathcal{Q}}$ is defined by*

$$k_{\mathcal{Q},t} = \begin{cases} \mathbf{k}, & (t \in \mathcal{Q}) \\ 0, & (t \notin \mathcal{Q}) \end{cases} \quad (5)$$

$$k_{\mathcal{Q}}(s \leq t) = \begin{cases} Id_{\mathbf{k}}, & (s \text{ and } t \in \mathcal{Q}) \\ 0, & \text{else} \end{cases} \quad (6)$$

Definition 2.14 (Interval). *In the poset $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^2$, $\mathcal{Q} \subset \mathcal{X} \times \mathcal{Y}$ is an interval if it is convex and zigzag path-connected, i.e. if between any two points $p, q \in \mathcal{Q}$, there exists a finite zigzag path $p \leq p_1 \geq p_2 \leq \dots \geq p_n \leq q$ with $p_i \in \mathcal{Q}$ for all $i = 1, 2, \dots, n$.*

Definition 2.15 (Interval module). *An indicator module $k_{\mathcal{Q}}$ is an interval module if \mathcal{Q} is an interval.*

Unfortunately, while a direct-sum decomposition of \mathbb{V} into indecomposables still exists and is essentially unique [BCB20], the summands may no longer be interval modules. For instance, the module in Figure 2.2 is indecomposable yet not an interval module nor even an indicator module. However, we can decompose the rank invariant of a bimodule as in [BOO21], see Definition 2.18.

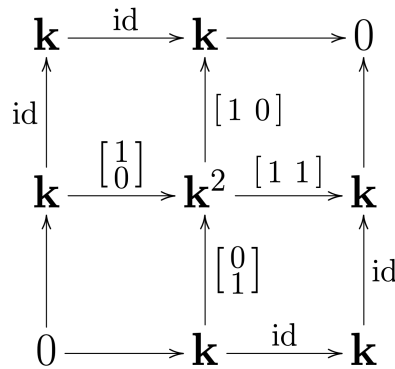


Figure 2.2: An indecomposable bimodule

Definition 2.16 (Rectangle). A set $R \subset \mathbb{R}^2$ is a rectangle if $R = I \times J$ where I and J are intervals in \mathbb{R} .

Definition 2.17 (Rectangle module). An indicator module k_Q is a rectangle module if Q is a rectangle.

Note that *rectangle modules* \subseteq *interval modules* \subseteq *indicator modules*. The only difference of these three types of persistence modules is the limitation for Q .

Definition 2.18 (Rank decomposition). Given a bimodule \mathbb{V} and its rank invariant $r = Rk\mathbb{V}$. A (signed) rank decomposition of r is given by the following kind of identity:

$$r = Rk\mathbf{k}_{\mathcal{R}} - Rk\mathbf{k}_{\mathcal{S}}$$

where \mathcal{R} and \mathcal{S} are multi-sets of rectangles and where $\mathbf{k}_{\mathcal{R}} = \bigoplus_{R \in \mathcal{R}} k_R$ and $\mathbf{k}_{\mathcal{S}} = \bigoplus_{S \in \mathcal{S}} k_S$. In other words, $Rk\mathbb{V}$ decomposes as a \mathbb{Z} -linear combination of rank invariants of rectangle modules, with possibly negative coefficients. By extension, we call the pair $(\mathcal{R}, \mathcal{S})$ itself a rank decomposition of r . It is a minimal rank decomposition if \mathcal{R} and \mathcal{S} are disjoint as multi-sets.

Theorem 2.3. Given a bimodule \mathbb{V} over $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^2$ and its rank invariant $r = Rk\mathbb{V}$, the minimal rank decomposition of r is unique when it exists. Moreover, it exists whenever $\mathcal{X} \times \mathcal{Y}$ is finite or \mathbb{V} is finitely presented. The multi-set of rectangles in the decomposition is called signed (persistence) barcode, denoted as $Dgm \mathbb{V}$.

Similar to single-parameter persistence, when $\mathcal{X} \times \mathcal{Y} \subset \mathbb{Z}^2$, the barcode $Dgm \mathbb{V}$ is easily obtained from $Rk\mathbb{V}$ via the following inclusion-exclusion formula [BLO22]:

$$\begin{aligned} m(s, t) = & r(s, t) - r((s_x - 1, s_y), t) \\ & - r((s_x, s_y - 1), t) + r((s_x - 1, s_y - 1), t), \\ & - r(s, (t_x + 1, t_y)) + r((s_x - 1, s_y), (t_x + 1, t_y)) \\ & + r((s_x, s_y - 1), (t_x + 1, t_y)) - r((s_x - 1, s_y - 1), (t_x + 1, t_y)) \\ & - r(s, (t_x, t_y + 1)) + r((s_x - 1, s_y), (t_x, t_y + 1)) \\ & + r((s_x, s_y - 1), (t_x, t_y + 1)) - r((s_x - 1, s_y - 1), (t_x, t_y + 1)) \\ & + r(s, (t_x + 1, t_y + 1)) - r((s_x - 1, s_y), (t_x + 1, t_y + 1)) \\ & - r((s_x, s_y - 1), (t_x + 1, t_y + 1)) + r((s_x - 1, s_y - 1), (t_x + 1, t_y + 1)) \end{aligned} \quad (7)$$

In order to apply this formula, we need to compute the rank invariant first, which is the main contribution of this report.

3 Algorithm

Based on Theorem 2.3 and formula (7), we develop an algorithm to compute the rank invariant then the signed rectangle decomposition of bimodules.

3.1 Description

Let \mathcal{F} be a simplicial bifiltration with n simplices. Assume that \mathcal{F} is indexed over the grid $G = [0, n_x - 1] \times [0, n_y - 1]$, and that each arrow $\mathcal{F}_{(i,j)} \rightarrow \mathcal{F}_{(i+1,j)}$ or $\mathcal{F}_{(i,j)} \rightarrow$

$\mathcal{F}_{(i,j+1)}$ is either an identity map or the insertion of some simplices. We fix a homology degree p .

Given the above input, here is the algorithm for the rank invariant of the persistence bimodule \mathbb{V} induced by \mathcal{F} in p -th homology. Consider all the paths of the form $(0,0) \rightarrow \dots \rightarrow (0,j) \rightarrow \dots \rightarrow (i,j) \rightarrow \dots \rightarrow (i,n_y-1) \rightarrow \dots \rightarrow (n_x-1,n_y-1)$ in the $n_x \times n_y$ grid, where $(i,j) \in [0, n_x-1] \times [0, n_y-1]$ is arbitrary, as shown in Figure 3.1(a) (the abscissae are represented vertically). We call such a path a *stair*, denoted by $S_{(i,j)}$, and the corresponding point (i,j) an *elbow*.

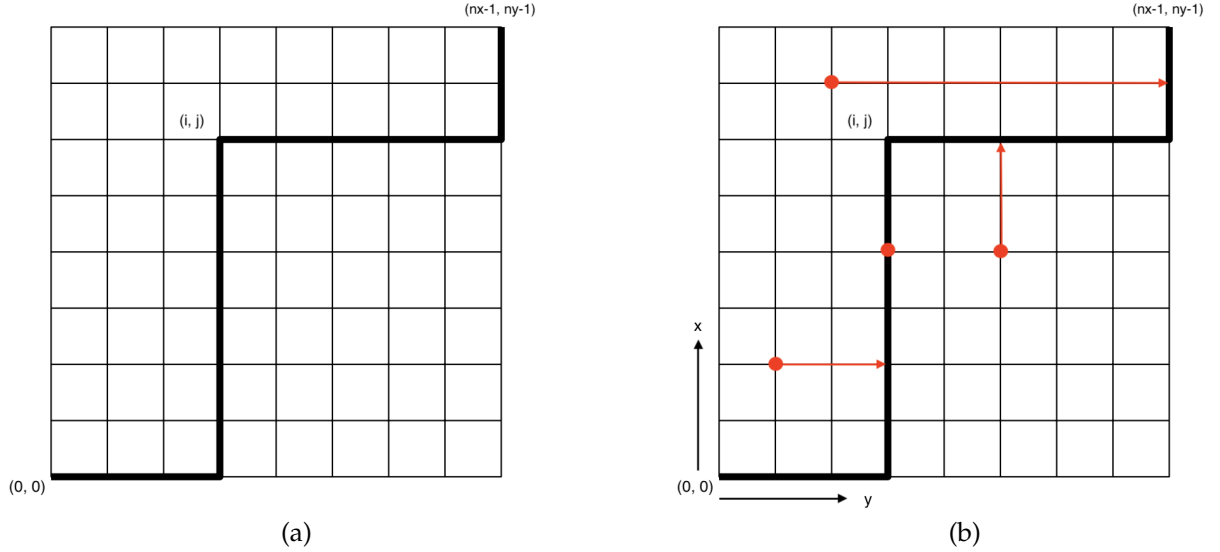


Figure 3.1: (a): The stair $S_{(i,j)}$. (b): The projection of points in the grid onto the path $S_{(i,j)}$. The abscissae are represented vertically. The red point means the insertion time of a simplex and the endpoint of the corresponding arrow is its projection onto the stair $S_{(i,j)}$.

We consider each path $S_{(i,j)}$ separately and compute the persistence barcode of the single-parameter restriction $\mathcal{F}|_{S_{(i,j)}}$.

Proposition 3.1. *Let \mathcal{F} be a simplicial bifiltration indexed over the grid $G = [0, n_x-1] \times [0, n_y-1]$ and r be the rank invariant of the persistence bimodule \mathbb{V} induced by \mathcal{F} in p -th homology. For any $(i,j) \in [0, n_x-1] \times [0, n_y-1]$ and any two points $s_1 = (x_1, y_1)$ and $s_2 = (x_2, y_2)$ in the stair $S_{(i,j)}$, the number of intervals in the persistence barcode passing through s_1 and s_2 is the value $r(s_1, s_2)$.*

Proof. Follows from Equation (1). □

The algorithm treats each path independently, but if there are two different paths passing through s_1 and s_2 , the values $r(s_1, s_2)$ from these two different single-parameter filtrations are the same because of the commutativity of the bimodule over the grid.

3.2 Implementation

The pseudocode is given in Algorithm 1. A simplex tree is used as the data structure to represent general simplicial complexes. The data structure is described in [BM14] and

implemented by GUDHI [The15]. For instance, Figure 3.2 shows a simplicial complex on 10 vertices. The deepest node of its simplex tree represents the tetrahedron of the complex. All the positions of a given label at a given depth are linked in a list, as illustrated in the case of label 5.

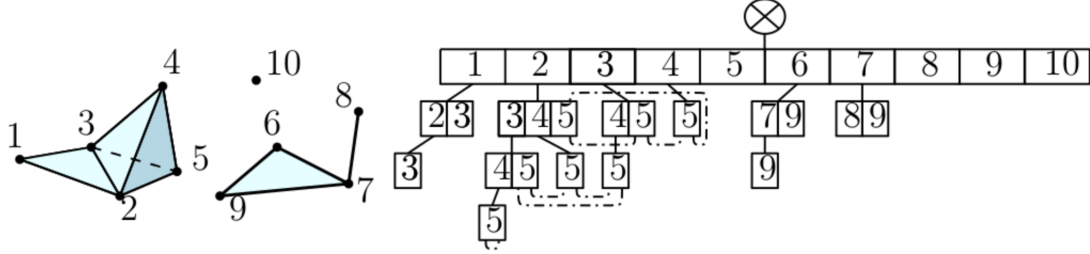


Figure 3.2: Simplex tree representation, from [The15].

With a simplex tree, we can insert simplices with filtration values, change filtration values of simplices, search a simplex and compute the persistent barcode of the simplicial complex.

The filtration value of a simplex in the simplex tree is assigned $x + y$, where (x, y) is the projection of the simplex insertion time (x', y') in the grid onto the path $S_{(i,j)}$. The projection is defined as the minimum point in $S_{(i,j)}$ that is greater than or equal to (x', y') in the product order, as illustrated in Figure 3.1(b).

Proposition 3.2 (Projection). *Let $S_{(i,j)}$ be a stair in the grid $G = [0, n_x - 1] \times [0, n_y - 1]$. The projection (x, y) of the point (x', y') onto the path $S_{(i,j)}$ is given by:*

$$(x, y) = \begin{cases} (x', y'), & (x', y') \in S_{(i,j)} \\ (x', j), & (x', y') \notin S_{(i,j)} \text{ and } x' \leq i \text{ and } y' \leq j \\ (i, y'), & (x', y') \notin S_{(i,j)} \text{ and } x' \leq i \text{ and } y' \geq j \\ (x', n_y - 1), & \text{otherwise} \end{cases}$$

For each path, we update the filtration values of all simplices instead of rebuilding the simplex tree, which saves a lot of time. And we do not search the simplex in the tree to update its filtration value but build a map from the simplex handle (the pointer to the simplex in the tree) to its insertion time in the grid before walking through all paths. Then for each path, we traverse the simplex tree to obtain the simplex handle of each simplex and its corresponding insertion time from the map. Finally, we can calculate the projection of the insertion time onto the path and update the filtration value of the simplex.

For one path, after updating the filtration values of all simplices, we can compute the persistence barcode of the single-parameter persistence. For each interval $[s, t]$ in the barcode, the points in the grid that correspond to the endpoints of the interval are the s -th and t -th points in $S_{(i,j)}$, so we add one to $r(S_{(i,j)}[s], S_{(i,j)}[t])$, by Proposition 3.2.

3.3 Complexity

The algorithm takes $O(n^\omega)$ per stair, using the single-parameter persistence algorithm based on fast matrix multiplication [MMS11], and as there are $O(n_x n_y)$ stairs in total, the overall running time of the algorithm is $O(n_x n_y n^\omega)$.

Algorithm 1 Compute the rank invariant for bimodule

Input: A simplicial bifiltration \mathcal{F} indexed over the grid $G = [0, n_x - 1] \times [0, n_y - 1]$
Define a simplex tree: st
Insert all simplices in \mathcal{F} into st with filtration value 0
Build a map from simplex in st to its born node $[x, y]$: $sgmap$
 $r \leftarrow zero_matrix(n_x, n_y, n_x, n_y)$
for $i = 0$ to $n_x - 1$ **do**
 for $j = 0$ to $n_y - 1$ **do**
 $elbow \leftarrow [i, j]$
 $stair \leftarrow [[0, 0], \dots, [i, 0], \dots, [i, j], \dots, [n_x - 1, j], \dots, [n_x - 1, n_y - 1]]$
 $r_temp \leftarrow zero_matrix(n_x, n_y, n_x, n_y)$
 for simplex in st **do**
 $node_grid \leftarrow sgmap[simplex]$
 $proj_p \leftarrow proj_grid_to_path(node_grid, stair, elbow)$
 $st.assign_filtration(simplex, proj_p.x + proj_p.y)$
 end for
 $barcodes \leftarrow st.compute_persistence()$
 for $[s, t)$ in $barcodes$ **do**
 for $ss = s$ to $\min\{t - 1, n_x + n_y - 2\}$ **do**
 for $tt = ss$ to $\min\{t - 1, n_x + n_y - 2\}$ **do**
 $r_temp(stair[ss], stair[tt]) \leftarrow r_temp(stair[ss], stair[tt]) + 1$
 end for
 end for
 end for
 end for
 for $s = [0, 0]$ to $[n_x - 1, n_y - 1]$ **do**
 for $t = [0, 0]$ to $[n_x - 1, n_y - 1]$ **do**
 if $r(s, t) = 0$ **then**
 $r(s, t) \leftarrow r_temp(s, t)$
 end if
 end for
 end for
end for
Output: rank invariant r

If the simplicial bifiltration \mathcal{F} has n simplices where $n \geq \max\{n_x, n_y\}$, then the space complexity is $O(nm)$, where m is the maximum dimension of faces in the simplicial complex.

4 Experimental Results

In this section, we use a toy example then some simulated data to verify the correctness of the algorithm. Then, we analyze its running time quantitatively on some larger datasets.

4.1 Correctness validation

4.1.1 Experiment 1: a toy example

The toy example is the bimodule shown in Figure 2.2, reproduced in Figure 4.1(a). The homology degree is 1 and $\mathbf{k} = \mathbb{Z}_2$.

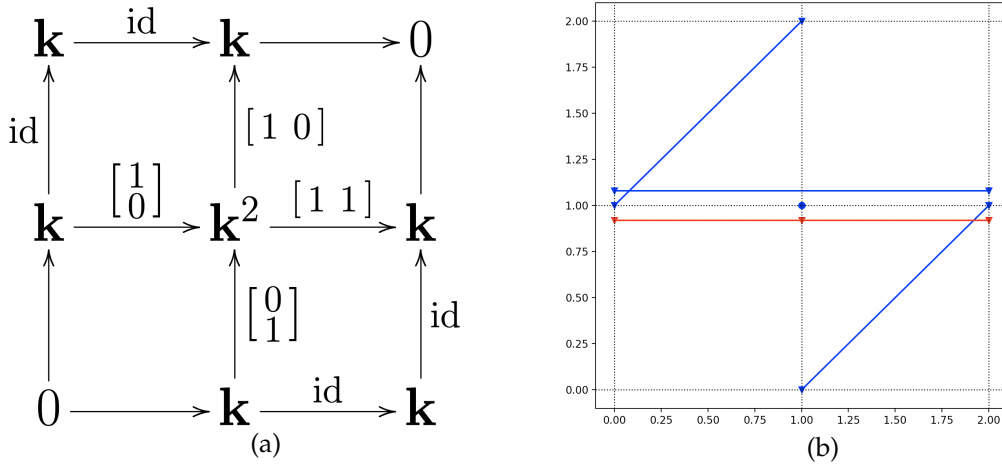


Figure 4.1: (a): a bimodule \mathbb{V} in the 3×3 grid. (b): the signed barcode corresponding to the rank decomposition of \mathbb{V} .

Its signed barcode decomposition is:

$$\begin{aligned}
 m((0,1), (1,2)) &= 1 \\
 m((0,1), (2,1)) &= 1 \\
 m((1,0), (2,1)) &= 1 \\
 m((1,1), (1,1)) &= 1 \\
 m((0,1), (1,1)) &= -1 \\
 m((1,1), (2,1)) &= -1
 \end{aligned}$$

which means:

$$\begin{aligned}
& Rk \left(\begin{array}{ccccc}
& \mathbf{k} & \xrightarrow{id} & \mathbf{k} & \longrightarrow 0 \\
& \uparrow id & & \uparrow [1 \ 0] & \uparrow \\
& \mathbf{k} & \longrightarrow & \mathbf{k}^2 & \longrightarrow \mathbf{k} \\
& \uparrow [0 \ 1] & & \uparrow [1 \ 1] & \uparrow id \\
0 & \longrightarrow & \mathbf{k} & \xrightarrow{id} & \mathbf{k}
\end{array} \right) = \\
& Rk \left(\begin{array}{cccc}
\begin{array}{ccccc}
0 & \longrightarrow & 0 & \longrightarrow & 0 \\
\uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & \mathbf{k} & \xrightarrow{id} & \mathbf{k} \\
\uparrow & & \uparrow id & & \uparrow id \\
0 & \longrightarrow & \mathbf{k} & \xrightarrow{id} & \mathbf{k}
\end{array} & \oplus & \begin{array}{ccccc}
0 & \longrightarrow & \mathbf{k} & \longrightarrow & 0 \\
\uparrow & & \uparrow id & & \uparrow \\
0 & \longrightarrow & \mathbf{k} & \longrightarrow & 0 \\
\uparrow & & \uparrow id & & \uparrow \\
0 & \longrightarrow & \mathbf{k} & \longrightarrow & 0
\end{array} & \oplus & \begin{array}{ccccc}
\mathbf{k} & \xrightarrow{id} & \mathbf{k} & \longrightarrow & 0 \\
\uparrow id & & \uparrow id & & \uparrow \\
\mathbf{k} & \xrightarrow{id} & \mathbf{k} & \longrightarrow & 0 \\
\uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & 0
\end{array} & \oplus & \begin{array}{ccccc}
0 & \longrightarrow & 0 & \longrightarrow & 0 \\
\uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & \mathbf{k} & \longrightarrow & 0 \\
\uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & 0
\end{array}
\end{array} \right) \\
& - Rk \left(\begin{array}{cccc}
\begin{array}{ccccc}
0 & \longrightarrow & 0 & \longrightarrow & 0 \\
\uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & \mathbf{k} & \longrightarrow & 0 \\
\uparrow & & \uparrow id & & \uparrow \\
0 & \longrightarrow & \mathbf{k} & \longrightarrow & 0
\end{array} & \oplus & \begin{array}{ccccc}
0 & \longrightarrow & \mathbf{k} & \longrightarrow & 0 \\
\uparrow & & \uparrow id & & \uparrow \\
0 & \longrightarrow & \mathbf{k} & \longrightarrow & 0 \\
\uparrow & & \uparrow & & \uparrow \\
0 & \longrightarrow & 0 & \longrightarrow & 0
\end{array}
\end{array} \right)
\end{aligned}$$

We can also visualize it as Figure 4.1(b). The blue segment between $s_1 = (0, 1)$ and $t_1 = (1, 2)$ means that $m(s_1, t_1)$ is positive and the red segment between $s_2 = (0, 1)$ and $t_2 = (2, 1)$ means that $m(s_2, t_2)$ is negative.

4.1.2 Experiment 2: bimodule clustering

The point set P consists of $N = 1000$ planar points sampled from four Gaussian distributions as shown in Figure 4.2(a). The so-called Rips bifiltration from the point set P has 200000 simplices. The homology degree is 0 and $\mathbf{k} = \mathbb{Z}_2$. The obtained signed barcode is shown in Figure 4.2(b).

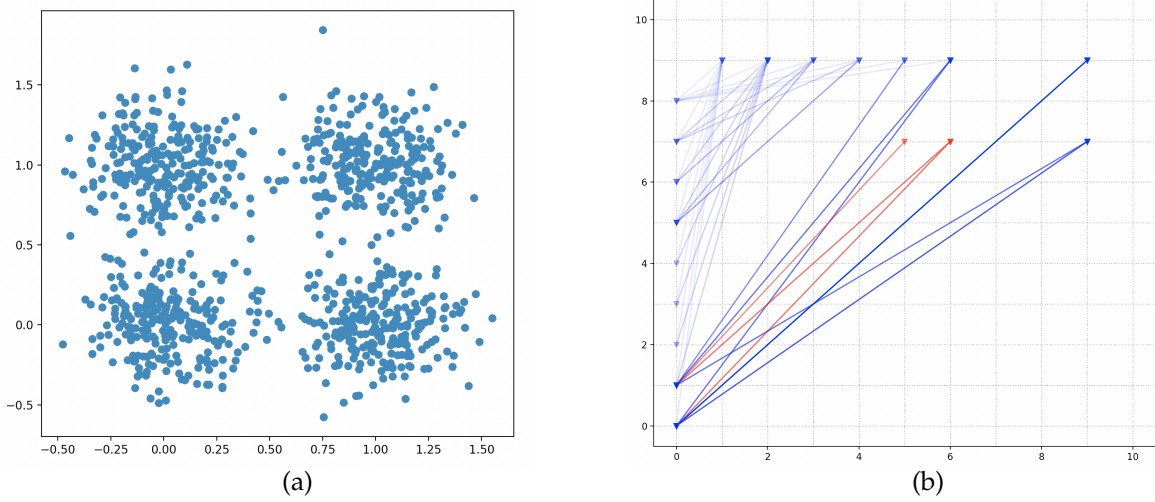


Figure 4.2: (a): Point cloud with 1000 points sampled from 4 Gaussian distributions (b): The corresponding signed barcode over a 10×10 grid. Each bar with endpoints $s \leq t$ in the barcode has an intensity proportional to $\min\{t_x - s_x, t_y - s_y\}$; in particular, horizontal and vertical bars are invisible

4.2 Running time

We test our algorithm on MacOS with 8GB memory and a 2.3 GHz Dual-Core Intel Core i5 processor. A single core is used. The running times of the various parts of the algorithm on one path are given in Table 4.2 and the total time is given in Table 4.1. We only calculate the signed barcode decomposition once, so we put it in Table 4.1. For 1 million simplices in a 20×20 grid, the algorithm takes about 1.2s per path. Table 4.3 shows the total time with different grid size and same number of simplices.

PART	SIMPLICES	GRID SIZE	TIME(s)
BARCODE DECOMPOSITION	1025	20×20	0.00038
	10007	20×20	0.00054
	100000	20×20	0.00045
	1000007	20×20	0.00054
TOTAL TIME	1025	20×20	0.31
	10007	20×20	2.58
	100000	20×20	35.32
	1000007	20×20	495.63

Table 4.1: Time for all paths: the total time seems to increase linearly with the number of simplices, while the time for calculating the barcode decomposition from the rank invariant remains mostly the same if the grid size does not change.

PART	SIMPLICES	GRID SIZE	TIME(s)
UPDATING A SIMPLEX TREE	1025	20×20	0.00022
	10007	20×20	0.0025
	100000	20×20	0.039
	1000007	20×20	0.58
CALCULATING BARCODES	1025	20×20	0.00032
	10007	20×20	0.0036
	100000	20×20	0.048
	1000007	20×20	0.64
CALCULATING RANK INVARIANT	1025	20×20	9.6E-05
	10007	20×20	0.00015
	100000	20×20	0.001
	1000007	20×20	0.012
TIME FOR ONE PATH	1025	20×20	0.00076
	10007	20×20	0.0064
	100000	20×20	0.088
	1000007	20×20	1.24

Table 4.2: Running time for one path: the running time of each part seems to increase linearly with the number of simplices.

SIMPLICES	GRID SIZE	TIME(s)
100000	10×5	4.19
100000	5×10	4.34
100000	10×10	8.29
100000	15×10	13.12
100000	10×15	12.83
100000	20×20	34.47

Table 4.3: Total time with different grid size: the total time seems to increase linearly with the size of grid.

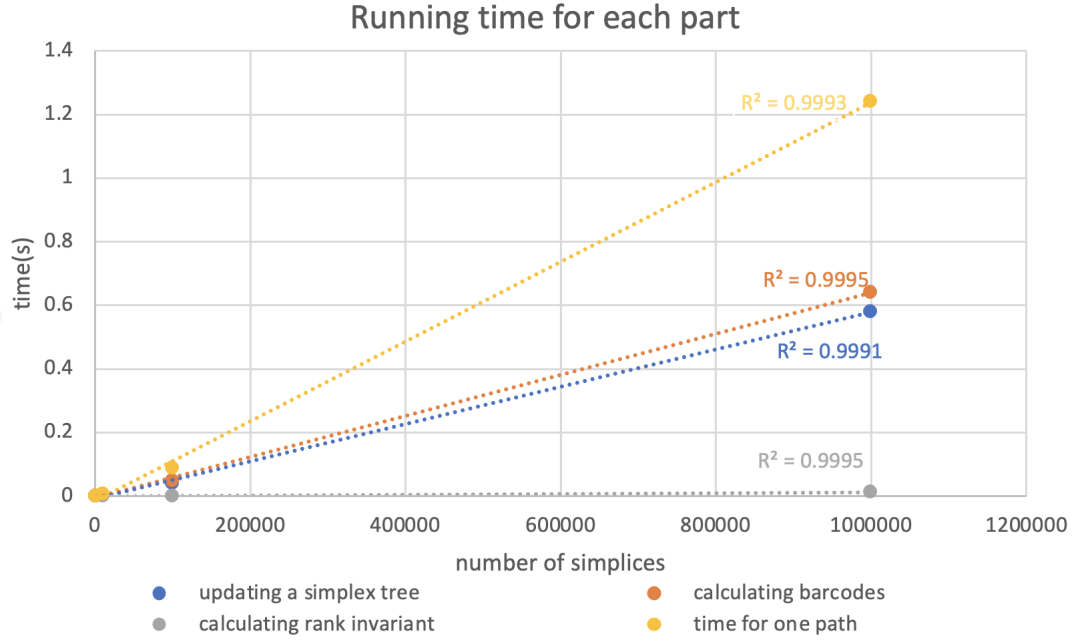


Figure 4.3: Curves for running time: linear regression is performed and the coefficient of determination is very closed to 1, so the running time of each part of the algorithm does increase linearly with the number of simplices.

From Table 4.2, we can see that the running time of each part seems to increase linearly with the number of simplices, so we perform linear regression as shown in Figure 4.3 to verify it. The four coefficients of determination are all very close to 1, so the running times of these four parts do increase linearly with the number of simplices. From Table 4.3, we can see that the total time also seems to increase linearly with the size of grid, which verifies the complexity of our algorithm.

References

- [BCB20] Magnus Botnan and William Crawley-Boevey. Decomposition of persistence modules. *Proceedings of the American Mathematical Society*, 148(11):4581–4596, 2020.
- [BLO22] Magnus Bakke Botnan, Vadim Lebovici, and Steve Oudot. On rectangle-decomposable 2-parameter persistence modules. *Discrete & Computational Geometry*, pages 1–24, 2022.
- [BM14] Jean-Daniel Boissonnat and Clément Maria. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 70(3):406–427, 2014.
- [BOO21] Magnus Bakke Botnan, Steffen Oppermann, and Steve Oudot. Signed barcodes for multi-parameter persistence via rank decompositions and rank-exact resolutions. *arXiv preprint arXiv:2107.06800*, 2021.
- [CDSGO16] Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. Springer, 2016.
- [CSEH07] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & computational geometry*, 37(1):103–120, 2007.
- [MMS11] Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the twenty-seventh Annual Symposium on Computational Geometry*, pages 216–225, 2011.

[The15] The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015.