Multi-parameter persistent homology: applications and algorithms

Nina Otter

Mathematical Institute, University of Oxford

Gudhi/Top Data Workshop

Porquerolles, 18 October 2016
Multi-parameter persistent homology pipeline

1. Data depending on $r$ parameters
2. Multi-filtered space
3. Multi-parameter persistence module
Step (1): from data to multi-filtered spaces

Define the following partial order on $\mathbb{N}^r$:

$$(u_1, \ldots, u_r) \leq (v_1, \ldots, v_r) \text{ iff } u_i \leq v_i \text{ for all } i = 1, \ldots, r.$$ 

A \textit{multi-filtered space} $K$ is a set of spaces $\{K_u\}_{u \in \mathbb{N}^r}$ such that $K_u \subseteq K_v$ if $u \leq v$ for all $u, v \in \mathbb{N}^r$. 

Map $f$: digital image with color vectors of length $r$ $\rightarrow$ $r$-filtered simplicial complex $\rightarrow$ $r$-filtered cubical complex.
Step (1): from data to multi-filtered spaces

Define the following partial order on $\mathbb{N}^r$:
$$(u_1, \ldots, u_r) \leq (v_1, \ldots, v_r) \iff u_i \leq v_i \text{ for all } i = 1, \ldots, r.$$ 

A *multi-filtered space* $K$ is a set of spaces $\{K_u\}_{u \in \mathbb{N}^r}$ such that $K_u \subseteq K_v$ if $u \leq v$ for all $u, v \in \mathbb{N}^r$.

Map $f : X \to \mathbb{R}^r$ $\to$ $r$-filtered simplicial complex

digital image with color vectors of length $r$ $\to$ $r$-filtered cubical complex
Step (1): from data to multi-filtered spaces: example
Step (2): from multi-filtered spaces to multi-parameter persistence modules

\[ r\text{-filtered space} \xrightarrow{H_i} r\text{-parameter persistence module} \]

An \textit{r-parameter persistence module} is a tuple \(( \{ M_i \}_{i \in \mathbb{N}^r}, \{ \phi_{i,j} \}_{i \leq j \in \mathbb{N}^r} )\) where:

- for each \( i \in \mathbb{N}^r \) we have that \( M_i \) is a \( k \)-module
- for every \( i \leq j \) we have that \( \phi_{i,j} : M_i \rightarrow M_j \) is a \( k \)-module homomorphism such that whenever \( i \leq k \leq j \) we have

\[ \phi_{k,j} \circ \phi_{i,k} = \phi_{i,j}. \]

In other words, an \textit{r-parameter persistence module} is a functor \( F : \mathbb{N}^r \rightarrow k\text{Mod}. \)
Interlude: representation theory of quivers

A quiver $Q = (Q_0, Q_1, s, t)$ consists of two non-empty sets $Q_0$ and $Q_1$ and two maps $s, t: Q_1 \to Q_0$. A quiver is finite if both $Q_0$ and $Q_1$ are finite.

Whenever $s(u) = x$ and $t(u) = y$ we write $x \xrightarrow{u} y$. For example, the following are finite quivers:

\[
\begin{array}{c}
x \xrightarrow{u} v \xrightarrow{w} y \\
\end{array}
\]
A quiver $Q = (Q_0, Q_1, s, t)$ consists of two non-empty sets $Q_0, Q_1$ and two maps $s, t: Q_1 \to Q_0$. A quiver is finite if both $Q_0$ and $Q_1$ are finite.

Whenever $s(u) = x$ and $t(u) = y$ we write $x \xrightarrow{u} y$. For example, the following are finite quivers:
Representations of quivers

Let $k$ be a field. A representation of a quiver $(V, \phi)$ consists of a family of $k$-vector spaces $V = \{V_i\}_{i \in Q_0}$ together with a family of $k$-linear maps $\phi = \{\phi_e : V_{s(e)} \to V_{t(e)} | e \in Q_1\}$. A representation $(V, \phi)$ is finite-dimensional if for all $i \in Q_0$ the vector space $V_i$ is finite-dimensional.

A morphism of representations $f : (V, \phi) \to (V', \phi')$ is given by $k$-linear maps $f_i : V_i \to V'_i$ for all $i \in Q_0$ such that the following diagram

\[
\begin{array}{ccc}
V_{s(e)} & \xrightarrow{\phi_e} & V_{t(e)} \\
\downarrow f_{s(e)} & & \downarrow f_{t(e)} \\
V'_{s(e)} & \xrightarrow{\phi'_t(e)} & V'_{t(e)}
\end{array}
\]

commutes for all $e \in Q_1$. 
Examples of quiver representations

Two finite-dimensional representations \( \phi: V' \to V \) and \( \psi: W' \to W \) are isomorphic iff 
\[
\dim V' = \dim W' \\
and \dim V = \dim W \\
and \text{rank} \phi = \text{rank} \psi.
\]

Two finite-dimensional representations \( \phi: V \to V \) and \( \psi: W \to W \) are isomorphic iff \( \phi \) and \( \psi \) have the same Jordan normal form.

Studying isomorphism classes of representations of this quiver amounts to studying pairs of quadratic matrices up to simultaneous conjugation.
Indecomposable representations

The direct sum of two representations \((\phi, V)\) and \((\psi, W)\) is the representation \((\phi \oplus \psi, V \oplus W)\) where \(V \oplus W = V_i \oplus W_i\) for all \(i \in Q_0\) and \((\phi \oplus \psi)_e = \begin{pmatrix} \phi_e & 0 \\ 0 & \psi_e \end{pmatrix}\).
Indecomposable representations

The *direct sum* of two representations \((\phi, V)\) and \((\psi, W)\) is the representation \((\phi \oplus \psi, V \oplus W)\) where \(V \oplus W = V_i \oplus W_i\) for all \(i \in Q_0\) and \((\phi \oplus \psi)_e = \begin{pmatrix} \phi_e & 0 \\ 0 & \psi_e \end{pmatrix}\).

We say that a representation \((\phi, V)\) is *indecomposable* if it is non-zero and not isomorphic to a direct sum of two non-zero representations.
Indecomposable representations

The *direct sum* of two representations \((\phi, V)\) and \((\psi, W)\) is the representation \((\phi \oplus \psi, V \oplus W)\) where \(V \oplus W = V_i \oplus W_i\) for all \(i \in Q_0\) and \((\phi \oplus \psi)_e = \begin{pmatrix} \phi_e & 0 \\ 0 & \psi_e \end{pmatrix}\).

We say that a representation \((\phi, V)\) is *indecomposable* if it is non-zero and not isomorphic to a direct sum of two non-zero representations.

**Example:** indecomposable representations of the loop quiver are given by the Jordan blocks.
Indecomposable representations

The direct sum of two representations \((\phi, V)\) and \((\psi, W)\) is the representation \((\phi \oplus \psi, V \oplus W)\) where \(V \oplus W = V_i \oplus W_i\) for all \(i \in Q_0\) and \((\phi \oplus \psi)_e = \begin{pmatrix} \phi_e & 0 \\ 0 & \psi_e \end{pmatrix}\).

We say that a representation \((\phi, V)\) is indecomposable if it is non-zero and not isomorphic to a direct sum of two non-zero representations.

Example: indecomposable representations of the loop quiver are given by the Jordan blocks.

Theorem (Krull, Remak, Schmidt) Assume that \(Q\) is finite, then any finite-dimensional representation \((V, \phi)\) of \(Q\) can be written as a direct sum \((V, \phi) = (V_1, \phi_1) \oplus \cdots \oplus (V_r, \phi_r)\) where each \((V_i, \phi_i)\) is indecomposable, and the decomposition is unique up to isomorphism and permutation of the terms.
Classification of (representations of) quivers

**Dynkin**

\[(n \geq 2)\]

\[
\begin{array}{cccc}
1 & 2 & \ldots & n-1 & n \\
\end{array}
\]

\[
\begin{array}{c}
\cdots \\
1 & 2 \\
\end{array}
\]

**Extended Dynkin**

\[(n \geq 2)\]

\[
\begin{array}{c}
0 \\
\end{array}
\]

\[
\begin{array}{cccc}
0 & 0 & 1 \\
\end{array}
\]

**Wild**

Everything else. For example:

\[
\begin{array}{c}
\infty \\
\end{array}
\]
Classification of representations of quivers

Suppose that $k$ is algebraically closed. The number of isomorphism classes of indecomposable representations is:

<table>
<thead>
<tr>
<th>Dynkin</th>
<th>Extended Dynkin</th>
<th>Wild</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite.</td>
<td>Infinite; depends on one parameter.</td>
<td>Infinite; depends on $N &gt; 1$ parameters, where $N$ depends on the quiver.</td>
</tr>
</tbody>
</table>

Classification of indecomposable representations of quivers: example

Consider again the loop quiver:
Classification of indecomposable representations of quivers: example

Consider again the loop quiver:

\[
\begin{array}{c}
\bullet \\
\end{array}
\]

Recall that two finite-dimensional representations \( \phi : V \to V \) and \( \psi : W \to W \) are isomorphic iff \( \phi \) and \( \psi \) have the same Jordan normal form, and the isomorphism classes of indecomposable representations of the loop quiver are given by the Jordan blocks.

Each Jordan block depends on a continuous parameter given by the eigenvalue.
Back to multi-parameter persistent homology

A multi-parameter persistence module is a representation of a quiver of the following form:

\[
\begin{array}{ccc}
\vdots & \vdots & \vdots \\
\uparrow & \uparrow & \uparrow \\
\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \cdots \\
\uparrow & \uparrow & \uparrow \\
\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \cdots \\
\uparrow & \uparrow & \uparrow \\
\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \cdots \\
\end{array}
\]
Back to multi-parameter persistent homology

A multi-parameter persistence module is a representation of a quiver of the following form:

```
... ... ...
\uparrow \quad \uparrow \quad \uparrow
\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \cdots
\uparrow \quad \uparrow \quad \uparrow
\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \cdots
\uparrow \quad \uparrow \quad \uparrow
\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \cdots
```

Such quivers are wild.
A multi-parameter persistence module is a representation of a quiver of the following form:

\[
\begin{array}{cccc}
\bullet & \rightarrow & \bullet & \rightarrow \cdots \\
\uparrow & & \uparrow & \\
\bullet & \rightarrow & \bullet & \rightarrow \cdots \\
\uparrow & & \uparrow & \\
\bullet & \rightarrow & \bullet & \rightarrow \cdots \\
\end{array}
\]

Such quivers are wild.

Our motivation/goal: find computable invariants for applications.
Application: Time evolution of blood vessel growth in presence of tumors

Roche, Oxford Oncology (B. Markelc), Mathematical Biology, University of Oxford (B. Stolz, H. Byrne, J. Grogan)
Persistence modules are modules

Recall that an $\mathbb{N}^r$-graded (or multi-graded) ring is a ring $A$ together with a collection $\{A^u\}_{u \in \mathbb{N}^r}$ of subgroups of the underlying abelian group of $A$ such that $A = \bigoplus_{u \in \mathbb{N}^r} A^u$ and for all $a \in A^m$ and $b \in A^n$ we have $ab \in A^{m+n}$.

Make the ring $A = k[x_1, \ldots, x_r]$ into an $\mathbb{N}^r$-graded ring by setting $A^u = kx_1^{u_1} \cdots x_r^{u_r}$ for all $u = (u_1, \ldots, u_r) \in \mathbb{N}^r$.

A module $M$ over an $\mathbb{N}^r$-graded ring $A$ is graded if there is a collection $\{M^i\}_{i \in \mathbb{N}^r}$ of subgroups of the underlying abelian group of $M$ such that $M = \bigoplus_{i \in \mathbb{N}^r} M^i$ and for all $a \in A^j$ we have $aM^i \subset M^{i+j}$.

Correspondence Theorem of Persistent Homology (Carlsson, Zomorodian '09)

The functor category of $r$-parameter persistence modules is isomorphic to the category of graded $k[x_1, \ldots, x_r]$-modules and module homomorphisms respecting the grading.
Persistence modules are modules

- Recall that an \( \mathbb{N}^r \)-graded (or multi-graded) ring is a ring \( A \) together with a collection \( \{ A_u \}_{u \in \mathbb{N}^r} \) of subgroups of the underlying abelian group of \( A \) such that \( A = \bigoplus_{u \in \mathbb{N}^r} A_u \) and for all \( a \in A_m \) and \( b \in A_n \) we have \( ab \in A_{m+n} \).
Persistence modules are modules

- Recall that an $\mathbb{N}^r$-graded (or multi-graded) ring is a ring $A$ together with a collection $\{A_u\}_{u \in \mathbb{N}^r}$ of subgroups of the underlying abelian group of $A$ such that $A = \bigoplus_{u \in \mathbb{N}^r} A_u$ and for all $a \in A_m$ and $b \in A_n$ we have $ab \in A_{m+n}$.

- Make the ring $A = k[x_1, \ldots, x_r]$ into an $\mathbb{N}^r$-graded ring by setting

$$A_u = kx_1^{u_1} \ldots x_r^{u_r} \text{ for all } u = (u_1, \ldots, u_r) \in \mathbb{N}^r.$$
Recall that an $\mathbb{N}^r$-graded (or multi-graded) ring is a ring $A$ together with a collection $\{A_u\}_{u \in \mathbb{N}^r}$ of subgroups of the underlying abelian group of $A$ such that $A = \bigoplus_{u \in \mathbb{N}^r} A_u$ and for all $a \in A_m$ and $b \in A_n$ we have $ab \in A_{m+n}$.

Make the ring $A = k[x_1, \ldots, x_r]$ into an $\mathbb{N}^r$-graded ring by setting

$$A_u = kx_1^{u_1} \ldots x_r^{u_r} \text{ for all } u = (u_1, \ldots, u_r) \in \mathbb{N}^r.$$ 

A module $M$ over an $\mathbb{N}^r$-graded ring $A$ is graded if there is a collection $\{M_i\}_{i \in \mathbb{N}^r}$ of subgroups of the underlying abelian group of $M$ such that $M = \bigoplus_{i \in \mathbb{N}^r} M_i$ and for all $a \in A_j$ we have $aM_i \subset M_{i+j}$. 

Correspondence Theorem of Persistent Homology (Carlsson, Zomorodian '09)
The functor category of $r$-parameter persistence modules is isomorphic to the category of graded $k[x_1, \ldots, x_r]$-modules and module homomorphisms respecting the grading.
Persistence modules are modules

- Recall that an $\mathbb{N}^r$-graded (or multi-graded) ring is a ring $A$ together with a collection $\{A_u\}_{u \in \mathbb{N}^r}$ of subgroups of the underlying abelian group of $A$ such that $A = \bigoplus_{u \in \mathbb{N}^r} A_u$ and for all $a \in A_m$ and $b \in A_n$ we have $ab \in A_{m+n}$.
- Make the ring $A = k[x_1, \ldots, x_r]$ into an $\mathbb{N}^r$-graded ring by setting

$$A_u = kx_1^{u_1} \cdots x_r^{u_r} \text{ for all } u = (u_1, \ldots, u_r) \in \mathbb{N}^r.$$  

- A module $M$ over an $\mathbb{N}^r$-graded ring $A$ is graded if there is a collection $\{M_i\}_{i \in \mathbb{N}^r}$ of subgroups of the underlying abelian group of $M$ such that $M = \bigoplus_{i \in \mathbb{N}^r} M_i$ and for all $a \in A_j$ we have $aM_i \subset M_{i+j}$.

Correspondence Theorem of Persistent Homology (Carlsson, Zomorodian '09)

The functor category of $r$-parameter persistence modules is isomorphic to the category of graded $k[x_1, \ldots, x_r]$-modules and module homomorphisms respecting the grading.
Any persistence module is the homology of a filtered space

The homology of a multi-filtered space is a persistence module.
Any persistence module is the homology of a filtered space

The homology of a multi-filtered space is a persistence module.

On the other hand:

**Theorem** (Carlsson, Zomorodian, 2009)
For any finite persistence module $M$ there exists a multi-filtered space $K$ and a positive natural number $i$ such that $M$ is the homology in degree $i$ of $K$. 
Any persistence module is the homology of a filtered space

The homology of a multi-filtered space is a persistence module.

On the other hand:

**Theorem** (Carlsson, Zomorodian, 2009)
For any finite persistence module $M$ there exists a multi-filtered space $K$ and a positive natural number $i$ such that $M$ is the homology in degree $i$ of $K$.

Therefore, studying the homology of $r$-filtered spaces amounts to studying graded modules over $k[x_1, \ldots, x_r]$. 
Free resolutions and presentations

Let $M$ be a finitely generated graded $k[x_1, \ldots, x_r]$-module. By the Hilbert Syzygy Theorem there is a free resolution by finitely generated $\mathbb{N}^r$-graded free $k[x_1, \ldots, x_r]$-modules of length at most $r$: 

$$0 \longrightarrow F_m \overset{\phi_m}{\longrightarrow} F_{m-1} \longrightarrow \cdots \longrightarrow F_1 \overset{\phi_1}{\longrightarrow} F_0 \longrightarrow M \longrightarrow 0$$

with image$(\phi_i) = \text{kernel}(\phi_{i-1})$ and each $F_i$ is a finitely generated graded free $k[x_1, \ldots, x_r]$-module and $m \leq r$. 
Free resolutions and presentations

Let $M$ be a finitely generated graded $k[x_1, \ldots, x_r]$-module. By the Hilbert Syzygy Theorem there is a free resolution by finitely generated $\mathbb{N}^r$-graded free $k[x_1, \ldots, x_r]$-modules of length at most $r$:

$$
0 \longrightarrow F_m \xrightarrow{\phi_m} F_{m-1} \longrightarrow \cdots \longrightarrow F_1 \xrightarrow{\phi_1} F_0 \longrightarrow M \longrightarrow 0
$$

with $\text{image}(\phi_i) = \text{kernel}(\phi_{i-1})$ and each $F_i$ is a finitely generated graded free $k[x_1, \ldots, x_r]$-module and $m \leq r$.

The first part

$$
F_1 \xrightarrow{\phi_1} F_0 \longrightarrow M \longrightarrow 0
$$

of a free resolution of a module is called *presentation*. If we are given a presentation of $M$, we can then explicitly write $M$ as the quotient $F_0/\text{im}\phi_1$. 
Minimal presentations and resolutions

Resolutions and presentations are in general not unique.

\[
M = (x_1 x_2, x_1 x_3) \subset k[x_1, x_2, x_3] = S.
\]

The following are two free resolutions of \(M\):

\[
0 \to S \xrightarrow{\begin{pmatrix} x_3 - x_2 \end{pmatrix}} S^2 (x_1 x_2 x_1 x_3) \to M \to 0
\]

\[
0 \xrightarrow{\begin{pmatrix} -x_1 \end{pmatrix}} S^2 \xrightarrow{\begin{pmatrix} x_3 x_2 x_3 - x_2^2 \end{pmatrix}} S^2 (x_1 x_2 x_1 x_3) \to M \to 0
\]

However, minimal presentations of modules over local or graded rings are unique up to isomorphism.
Minimal presentations and resolutions

Resolutions and presentations are in general not unique.

**Example:**¹ Let $M = (x_1 x_2, x_1 x_3) \subset k[x_1, x_2, x_3] = S$. The following are two free resolutions of $M$:

$$
\begin{align*}
0 & \longrightarrow S \\
& \quad \longrightarrow S^2 \\
& \quad \longrightarrow M \\
& \quad \longrightarrow 0
\end{align*}
$$

$$
\begin{align*}
0 & \longrightarrow S \\
& \quad \longrightarrow S^2 \\
& \quad \longrightarrow S^2 \\
& \quad \longrightarrow M \\
& \quad \longrightarrow 0
\end{align*}
$$

However, minimal presentations of modules over local or graded rings are unique up to isomorphism.

¹Bulletin of the AMS, July 2016
Minimal presentations are invariants of a module, and one can compute many invariants from minimal presentations and resolutions, such as:

- Betti numbers
- (Multi-graded) Hilbert series
- ...
Presentation of a persistence module: naïve Algorithm

Since the $i$th homology of the $i$th chain complex of a multi-filtered simplicial complex is defined as

$$H_i = \text{kernel}(d_i)/\text{image}(d_{i+1}),$$

an algorithm to compute a presentation of $H_i$ is given by the following steps:

1. Compute a presentation of $\text{image}(d_{i+1})$.

---

Presentation of a persistence module: naïve Algorithm

Since the $i$th homology of the $i$th chain complex of a multi-filtered simplicial complex is defined as

$$H_i = \text{kernel}(d_i) / \text{image}(d_{i+1}),$$

an algorithm to compute a presentation of $H_i$ is given by the following steps:

1. Compute a presentation of $\text{image}(d_{i+1})$.
2. Compute a presentation of $\text{kernel}(d_i)$.

---

Presentation of a persistence module: naïve Algorithm

Since the $i$th homology of the $i$th chain complex of a multi-filtered simplicial complex is defined as

$$H_i = \text{kernel}(d_i) / \text{image}(d_{i+1}),$$

an algorithm to compute a presentation of $H_i$ is given by the following steps:

1. Compute a presentation of $\text{image}(d_{i+1})$.
2. Compute a presentation of $\text{kernel}(d_i)$.
3. Compute a presentation of the quotient $H_i$.

**Problem:** the known algorithms to compute $\text{image}(d_{i+1})$ are exponential in time and space$^1$.

---

Presentation of a module: algorithm by Carlsson, Singh and Zomorodian

In 2010 Carlsson, Singh and Zomorodian\textsuperscript{2} put forward an optimisation of the algorithm to compute a presentation of homology of ‘one-critical’ multi-filtered complexes.

Presentation of a module: algorithm by Carlsson, Singh and Zomorodian

In 2010 Carlsson, Singh and Zomorodian\(^2\) put forward an optimisation of the algorithm to compute a presentation of homology of ‘one-critical’ multi-filtered complexes.

Let \( K = \{ K_u \}_{u \in \mathbb{N}^r} \) be a multi-filtered simplicial complex. We assume that there exists \( v \in \mathbb{N}^r \) such that \( K_v \) is a finite simplicial complex, and \( K_u = K_v \) for all \( u \geq v \). We denote the simplicial complex \( K_v \) by \( K_{\text{tot}} \).

In 2010 Carlsson, Singh and Zomorodian\textsuperscript{2} put forward an optimisation of the algorithm to compute a presentation of homology of ‘one-critical’ multi-filtered complexes.

Let $K = \{K_u\}_{u \in \mathbb{N}^r}$ be a multi-filtered simplicial complex. We assume that there exists $v \in \mathbb{N}^r$ such that $K_v$ is a finite simplicial complex, and $K_u = K_v$ for all $u \geq v$. We denote the simplicial complex $K_v$ by $K_{\text{tot}}$.

For any $\sigma \in K_{\text{tot}}$ define the set of \textit{generators of $\sigma$} to be

$$\text{gen}(\sigma) = \min\{v \in \mathbb{N}^r \mid \sigma \in K(v)\}.$$
Presentation of a module: algorithm by Carlsson, Singh and Zomorodian

In 2010 Carlsson, Singh and Zomorodian\textsuperscript{2} put forward an optimisation of the algorithm to compute a presentation of homology of ‘one-critical’ multi-filtered complexes.

Let $K = \{K_u\}_{u \in \mathbb{N}^r}$ be a multi-filtered simplicial complex. We assume that there exists $v \in \mathbb{N}^r$ such that $K_v$ is a finite simplicial complex, and $K_u = K_v$ for all $u \geq v$. We denote the simplicial complex $K_v$ by $K_{\text{tot}}$.

For any $\sigma \in K_{\text{tot}}$ define the set of generators of $\sigma$ to be

$$\text{gen}(\sigma) = \min\{v \in \mathbb{N}^r \mid \sigma \in K(v)\}.$$

A multi-filtered simplicial complex is \textit{one-critical} if the set of generators of every simplex has cardinality one.

One-critical multi-filtered simplicial complex: example

One-critical:

(0, 1)  (1, 1)

(0, 0)  (1, 0)
One-critical multi-filtered simplicial complex: example

One-critical:

- \((0, 1)\)
- \((1, 1)\)
- \((0, 0)\)
- \((1, 0)\)

Not one-critical:

- \((0, 1)\)
- \((1, 1)\)
- \((0, 0)\)
- \((1, 0)\)
Optimization

For a one-critical multifiltered simplicial complex $K$:

- the chain modules are free modules, hence one can choose bases for them
- The standard basis is the basis of simplices in degree given by their generator
- The boundary maps can be written as homogeneous matrices with monomial entries
- Carlsson, Singh and Zomorodian show that this gives a polynomial bound on complexity.
- The resulting presentation is not an invariant, as it depends on a choice of basis.
In 2014 Chacholski, Scolamiero and Vaccarino\textsuperscript{3} put forward a polynomial-time algorithm to compute a presentation of homology of arbitrary multi-filtered simplicial complexes.

\textsuperscript{3}W. Chacholski, M. Scolamiero, and F. Vaccarino, \textit{Combinatorial presentation of multidimensional persistent homology}, 2014.
Presentation of a module: algorithm by Chacholski-Scolamiero-Vaccarino (CSV)

In 2014 Chacholski, Scolamiero and Vaccarino\(^3\) put forward a polynomial-time algorithm to compute a presentation of homology of arbitrary multi-filtered simplicial complexes.

\[\begin{itemize}
\item For any \( u \in \mathbb{N}^r \), denote by \( K_{n,u} \) the set of \( n \)-simplices in \( K_u \); the assignment \( u \mapsto K_{n,u} \) induces a functor \( K_n : \mathbb{N}^r \to \text{Sets} \), where \( \text{Sets} \) is the category of sets.
\item For any \( v \in \mathbb{N}^r \) and any \( i \in \{0, \ldots, n+1\} \) define the following map
\[ d_i : K_{n+1,v} \longrightarrow K_{n,v} : \{x_0, \ldots, x_{n+1}\} \mapsto \{x_0, \ldots, \hat{x}_i, \ldots, x_{n+1}\} \]
where \( \hat{x}_i \) means that we omit the vertex \( x_i \). The maps \( d_i \) give natural transformations \( K_{n+1} \to K_n \).
\end{itemize}\]

Let $S = k[x_1, \ldots, x_r]$. There exists a sequence of free graded $S$-modules

$$
\mathcal{RK}_n \oplus \mathcal{RG}_{K_{n+1}} \xrightarrow{\pi \oplus d} \mathcal{RG}_K \xrightarrow{\alpha} \mathcal{RD}_{n-1}
$$

such that $\alpha \circ (\pi \oplus d)$ is trivial, and the $k[x_1, \ldots, x_r]$-module $\text{kernel}(\alpha)/\text{im}(\pi \oplus d)$ is isomorphic to the homology in degree $n$ of the multi-filtered simplicial complex $K$. 


CSV algorithm

Let $S = k[x_1, \ldots, x_r]$. There exists a sequence of free graded $S$-modules

$$
\mathcal{RK}_n \oplus \mathcal{RG}_{K_n+1} \xrightarrow{\pi \oplus d} \mathcal{RG}_n \xrightarrow{\alpha} \mathcal{RD}_{n-1}
$$

such that $\alpha \circ (\pi \oplus d)$ is trivial, and the $k[x_1, \ldots, x_r]$-module $\ker(\alpha)/\text{im}(\pi \oplus d)$ is isomorphic to the homology in degree $n$ of the multi-filtered simplicial complex $K$, where

- $\mathcal{RK}_n = \bigoplus_{\sigma \in K_{\text{tot}}, n} \bigoplus_{v_0 \neq v_1 \in \text{gen}(\sigma)} x^{\max\{v_0, v_1\}} S$,
- $\mathcal{RG}_n = \bigoplus_{\sigma \in K_{\text{tot}}, n} \bigoplus_{v \in \text{gen}(\sigma)} x^v S$, and
- $\mathcal{RD}_{n-1} = \bigoplus_{\sigma \in K_{\text{tot}}, n-1} S$.

with the notation $x^u := x_1^{u_1} \ldots x_r^{u_r}$. 
CSV algorithm

The homomorphisms are defined as follows:

\(\pi\) For any \(\sigma \in K_n\) and \(v_0 \neq v_1 \in \text{gen}(\sigma)\), the homomorphism \(\pi : \mathcal{R}K_n \to \mathcal{R}G K_n\) sends \(x^{\max\{v_0, v_1\}}\) to \(x^{v_0} - x^{v_1}\).

\(d\) For any \(\sigma \in K_{n+1}\) and \(v \in \text{gen}(\sigma)\), the homomorphism \(d : \mathcal{R}G K_{n+1} \to \mathcal{R}G K_n\) sends \(x^v\) to \(\sum_{i=0}^{n+1} (-1)^i x^{\tilde{d}_i(\sigma)}\), where \(\tilde{d}_i(\sigma)\) is the minimal element in the set \(\{w \in \text{gen}(d_i(\sigma)) \mid w \leq v\}\) with respect to the lexicographical order.

\(\alpha\) For any \(\sigma \in K_n\) and \(v \in \text{gen}(\sigma)\), the homomorphism \(\alpha : \mathcal{R}G K_n \to \mathcal{R}D_{n-1}\) sends \(x^v\) to \(\sum_{i=0}^{n} (-1)^i d_i(\sigma)\).
CSV algorithm: example

We illustrate the algorithm for the computation of $H_1$ of the following 2-filtered simplicial complex $K$:

We have:

$\text{gen}(\{a\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{b\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{c\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{a, b\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{b, c\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{a, c\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{a, b, c\}) = \{(1,1)\}$
CSV algorithm: example

We illustrate the algorithm for the computation of $H_1$ of the following 2-filtered simplicial complex $K$:

We have:

$$\text{gen}\{\{a\}\} = \{(0, 1), (1, 0)\}$$
CSV algorithm: example

We illustrate the algorithm for the computation of $H_1$ of the following 2-filtered simplicial complex $K$:

We have:

$\text{gen}(\{a\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{b\}) = \{(0,1), (1,0)\}$
We illustrate the algorithm for the computation of $H_1$ of the following 2-filtered simplicial complex $K$:

We have:

$\text{gen}\{\{a\}\} = \{(0,1),(1,0)\}$
$\text{gen}\{\{b\}\} = \{(0,1),(1,0)\}$
$\text{gen}\{\{c\}\} = \{(0,1),(1,0)\}$
CSV algorithm: example

We illustrate the algorithm for the computation of $H_1$ of the following 2-filtered simplicial complex $K$:

We have:

$$\text{gen}(\{a\}) = \{(0, 1), (1, 0)\}$$
$$\text{gen}(\{b\}) = \{(0, 1), (1, 0)\}$$
$$\text{gen}(\{c\}) = \{(0, 1), (1, 0)\}$$
$$\text{gen}(\{a, b\}) = \{(0, 1), (1, 0)\}$$
CSV algorithm: example

We illustrate the algorithm for the computation of $H_1$ of the following 2-filtered simplicial complex $K$:

We have:

$\text{gen}(\{a\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{b\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{c\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{a, b\}) = \{(0,1), (1,0)\}$

$\text{gen}(\{b, c\}) = \{(0,1), (1,0)\}$
CSV algorithm: example

We illustrate the algorithm for the computation of $H_1$ of the following 2-filtered simplicial complex $K$:

We have:

- $\text{gen}(\{a\}) = \{(0, 1), (1, 0)\}$
- $\text{gen}(\{b\}) = \{(0, 1), (1, 0)\}$
- $\text{gen}(\{c\}) = \{(0, 1), (1, 0)\}$
- $\text{gen}(\{a, b\}) = \{(0, 1), (1, 0)\}$
- $\text{gen}(\{b, c\}) = \{(0, 1), (1, 0)\}$
- $\text{gen}(\{a, c\}) = \{(0, 1), (1, 0)\}$
CSV algorithm: example

We illustrate the algorithm for the computation of $H_1$ of the following 2-filtered simplicial complex $K$:

We have:

\[
\begin{align*}
\text{gen}(\{a\}) &= \{(0,1),(1,0)\} \\
\text{gen}(\{b\}) &= \{(0,1),(1,0)\} \\
\text{gen}(\{c\}) &= \{(0,1),(1,0)\} \\
\text{gen}(\{a, b\}) &= \{(0,1),(1,0)\} \\
\text{gen}(\{b, c\}) &= \{(0,1),(1,0)\} \\
\text{gen}(\{a, c\}) &= \{(0,1),(1,0)\} \\
\text{gen}(\{a, b, c\}) &= \{(1,1)\}
\end{align*}
\]
CSV algorithm: example

Let $S = k[x_1, x_2]$. Then:

$\mathcal{RK}K_1 = \bigoplus_{\sigma \in \mathcal{K}_{tot}, 1} \bigoplus_{v_0 \neq v_1 \in \text{gen}(\sigma)} x^{\max\{v_0, v_1\}} S$
CSV algorithm: example

Let $S = k[x_1, x_2]$. Then:

$\mathcal{RK}_1 = \bigoplus_{\sigma \in \text{tot}, 1} \bigoplus_{v_0 \neq v_1 \in \text{gen}(\sigma)} x^{\max\{v_0, v_1\}} S$, so

$$\mathcal{RK}_1 = x_1 x_2 S \oplus x_1 x_2 S \oplus x_1 x_2 S$$
CSV algorithm: example

Let $S = k[x_1, x_2]$. Then:

- $\mathcal{RK} K_1 = \bigoplus_{\sigma \in K_{tot,1}} \bigoplus_{v_0 \neq v_1 \in \text{gen}(\sigma)} x^{\max\{v_0, v_1\}} S$, so

  $\mathcal{RK} K_1 = x_1 x_2 S \oplus x_1 x_2 S \oplus x_1 x_2 S$

- $\mathcal{RG} K_n = \bigoplus_{\sigma \in K_{tot,n}} \bigoplus_{v \in \text{gen}(\sigma)} x^v S$
CSV algorithm: example

Let $S = k[x_1, x_2]$. Then:

- $\mathcal{RK}K_1 = \bigoplus_{\sigma \in K_{\text{tot}}, 1} \bigoplus_{v_0 \neq v_1 \in \text{gen}(\sigma)} x^{\max\{v_0, v_1\}} S$, so $\mathcal{RK}K_1 = x_1x_2 S \oplus x_1x_2 S \oplus x_1x_2 S$

- $\mathcal{RG}K_n = \bigoplus_{\sigma \in K_{\text{tot}}, n} \bigoplus_{v \in \text{gen}(\sigma)} x^v S$, so $\mathcal{RG}K_2 = x_1x_2 S$
Let $S = k[x_1, x_2]$. Then:

$\mathcal{R}_K K_1 = \bigoplus_{\sigma \in K_{\text{tot}, 1}} \bigoplus_{v_0 \neq v_1 \in \text{gen}(\sigma)} x^{\max\{v_0, v_1\}} S$, so

$\mathcal{R}_K K_1 = x_1 x_2 S \oplus x_1 x_2 S \oplus x_1 x_2 S$

$\mathcal{R}_G K_n = \bigoplus_{\sigma \in K_{\text{tot}, n}} \bigoplus_{v \in \text{gen}(\sigma)} x^v S$, so

$\mathcal{R}_G K_2 = x_1 x_2 S$

$\mathcal{R}_G K_1 = x_1 S \oplus x_2 S \oplus x_1 S \oplus x_2 S \oplus x_1 S \oplus x_2 S$
Let $S = k[x_1, x_2]$. Then:

- $\mathcal{RK} K_1 = \bigoplus_{\sigma \in K_{\text{tot}, 1}} \bigoplus_{\nu_0 \neq \nu_1 \in \text{gen} (\sigma)} x^{\max \{\nu_0, \nu_1\}} S$, so
  $$\mathcal{RK} K_1 = x_1 x_2 S \oplus x_1 x_2 S \oplus x_1 x_2 S$$

- $\mathcal{RG} K_n = \bigoplus_{\sigma \in K_{\text{tot}, n}} \bigoplus_{\nu \in \text{gen} (\sigma)} x^{\nu} S$, so
  $$\mathcal{RG} K_2 = x_1 x_2 S$$
  $$\mathcal{RG} K_1 = x_1 S \oplus x_2 S \oplus x_1 S \oplus x_2 S \oplus x_1 S \oplus x_2 S$$

- $\mathcal{RD}_0 = \bigoplus_{\sigma \in K_{\text{tot}, 0}} k[x_1, \ldots, x_r]$, so
Let $S = k[x_1, x_2]$. Then:

$\mathcal{R}K_1 = \bigoplus_{\sigma \in \mathcal{K}_{\text{tot}}, 1} \bigoplus_{v_0 \neq v_1 \in \text{gen}(\sigma)} x^{\max\{v_0, v_1\}} S$, so

$\mathcal{R}K_1 = x_1 x_2 S \oplus x_1 x_2 S \oplus x_1 x_2 S$

$\mathcal{R}G K_n = \bigoplus_{\sigma \in \mathcal{K}_{\text{tot}}, n} \bigoplus_{v \in \text{gen}(\sigma)} x^v S$, so

$\mathcal{R}G K_2 = x_1 x_2 S$

$\mathcal{R}G K_1 = x_1 S \oplus x_2 S \oplus x_1 S \oplus x_2 S \oplus x_1 S \oplus x_2 S$

$\mathcal{R}D_0 = \bigoplus_{\sigma \in \mathcal{K}_{\text{tot}}, 0} k[x_1, \ldots, x_r]$, so

$\mathcal{R}DK_0 = S \oplus S \oplus S$
CSV algorithm: example

\[\pi: RK_K_1 \rightarrow RG_K_1\] sends \(x^\max\{v_0, v_1\}\) to \(x^{v_0} - x^{v_1}\), so
CSV algorithm: example

\[ \pi: RKK_1 \rightarrow RGK_1 \text{ sends } x^{\max\{v_0, v_1\}} \text{ to } x^{v_0} - x^{v_1}, \text{ so} \]

\[
\pi = \begin{pmatrix}
x_1 & 0 & 0 \\
-x_2 & 0 & 0 \\
0 & x_1 & 0 \\
0 & -x_2 & 0 \\
0 & 0 & x_1 \\
0 & 0 & -x_2
\end{pmatrix}
\]
CSV algorithm: example

- $\pi: RK_1 \rightarrow RG_1$ sends $x^{\max\{v_0, v_1\}}$ to $x^{v_0} - x^{v_1}$, so
  \[
  \pi = \begin{pmatrix}
  x_1 & 0 & 0 \\
  -x_2 & 0 & 0 \\
  0 & x_1 & 0 \\
  0 & -x_2 & 0 \\
  0 & 0 & x_1 \\
  0 & 0 & -x_2
  \end{pmatrix}
  \]

- $d: RG_2 \rightarrow RG_1$ sends $x^\nu$ to $\sum_{i=0}^{n+1} (-1)^i x^{\overline{d_i(\sigma)}}$
CSV algorithm: example

- $\pi: R^KK_1 \rightarrow RGK_1$ sends $x^{\max\{v_0, v_1\}}$ to $x^{v_0} - x^{v_1}$, so

\[
\pi = \begin{pmatrix}
  x_1 & 0 & 0 \\
  -x_2 & 0 & 0 \\
  0 & x_1 & 0 \\
  0 & -x_2 & 0 \\
  0 & 0 & x_1 \\
  0 & 0 & -x_2 
\end{pmatrix}
\]

- $d: RGK_2 \rightarrow RGK_1$ sends $x^v$ to $\sum_{i=0}^{n+1} (-1)^i x^{d_i(\sigma)}$, so

\[
d = (0 \quad -x_2 \quad 0 \quad -x_2 \quad 0 \quad -x_2)^t
\]
CSV algorithm: example

- $\pi: RK_{K_1} \rightarrow RG_{K_1}$ sends $x^{\max\{v_0, v_1\}}$ to $x^{v_0} - x^{v_1}$, so
  \[\pi = \begin{pmatrix} x_1 & 0 & 0 \\ -x_2 & 0 & 0 \\ 0 & x_1 & 0 \\ 0 & -x_2 & 0 \\ 0 & 0 & x_1 \\ 0 & 0 & -x_2 \end{pmatrix}\]

- $d: RG_{K_2} \rightarrow RG_{K_1}$ sends $x^v$ to $\sum_{i=0}^{n+1} (-1)^i x^{\overline{d_i(\sigma)}}$, so
  \[d = \begin{pmatrix} 0 & -x_2 & 0 & -x_2 & 0 & -x_2 \end{pmatrix}^t\]

- $\alpha: RG_{K_1} \rightarrow RD_0$ sends $x^v$ to $\sum_{i=0}^{n} (-1)^i d_i(\sigma)$
CSV algorithm: example

- \( \pi: RK_{K_1} \rightarrow RG_{K_1} \) sends \( x^{\max\{v_0, v_1\}} \) to \( x^{v_0} - x^{v_1} \), so

\[
\pi = \begin{pmatrix}
x_1 & 0 & 0 \\
-x_2 & 0 & 0 \\
0 & x_1 & 0 \\
0 & -x_2 & 0 \\
0 & 0 & x_1 \\
0 & 0 & -x_2
\end{pmatrix}
\]

- \( d: RG_{K_2} \rightarrow RG_{K_1} \) sends \( x^{v} \) to \( \sum_{i=0}^{n+1} (-1)^i x^{d_i(\sigma)} \), so

\[
d = (0 \quad -x_2 \quad 0 \quad -x_2 \quad 0 \quad -x_2)^t
\]

- \( \alpha: RG_{K_1} \rightarrow RD_0 \) sends \( x^{v} \) to \( \sum_{i=0}^{n} (-1)^i d_i(\sigma) \), so

\[
\alpha = \begin{pmatrix}
-1 & -1 & -1 & -1 & 0 & 0 \\
1 & 1 & 0 & 0 & -1 & -1 \\
0 & 0 & 1 & 1 & 1 & 1
\end{pmatrix}
\]
By using a computational algebra software package one can then compute the following minimal presentation:

\[
0 \rightarrow S^2 \xrightarrow{\begin{pmatrix} x_1 & 0 \\ 0 & x_2 \end{pmatrix}} S^2 \xrightarrow{(x_2 \ x_1)} H_1(K) \rightarrow 0
\]

and thus

\[
H_1(K) = \frac{x_1 S}{(x_1 x_2)} \oplus \frac{x_2 S}{(x_1 x_2)}.
\]
CSV algorithm: example

\[ H_1(K) = \frac{x_1 S}{(x_1 x_2)} \oplus \frac{x_2 S}{(x_1 x_2)}. \]
Conclusions

- Need efficient implementation of algorithm by Chacholski, Scolamiero and Vaccarino.
- Computational algebra libraries are not efficient.
- How complex is the problem in practice?
- Insight from geometric invariant theory?
Conclusions

- Need efficient implementation of algorithm by Chacholski, Scolamiero and Vaccarino.

- Computational algebra libraries are not efficient.

- How complex is the problem in practice?

- Insight from geometric invariant theory?