Towards an Optimal Bound for the Interleaving Distance on Mapper Graphs

Ishika Ghosh

with Erin Wolf Chambers, Elizabeth Munch, Sarah Percival, Bei Wang

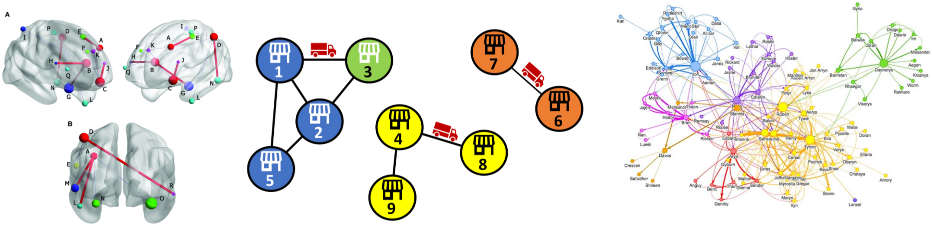
Department of Computational Mathematics, Science and Engineering
Michigan State University

Topological Data Visualization Workshop 2025

June 9, 2025

Graphs in the Wild

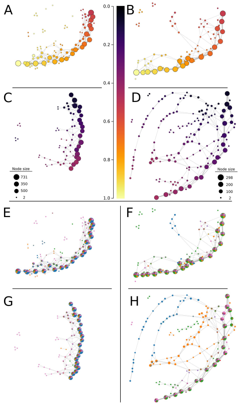
- Graphs with functions show up everywhere.
- We want to cluster and compare them.
- Need a meaningful and computable distance.
- But it's expensive to compute.



Real-world examples: brain activity, traffic flow, and social networks

Key Problem

- Study graph-based topological signatures.
- Focus on Mapper graphs.
- Use interleaving distance.
- Computation is NP-hard.
- Loss function to upper bound.
- How to get the best upper bound?



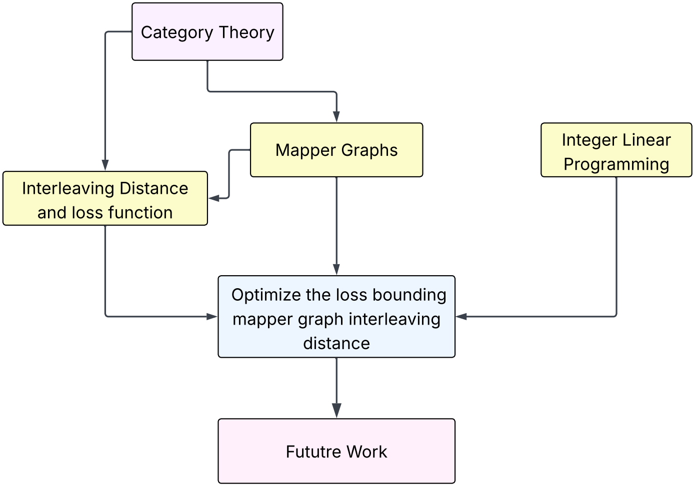
Mapper graphs of leaf data

Image credit: Percival et al., 2024

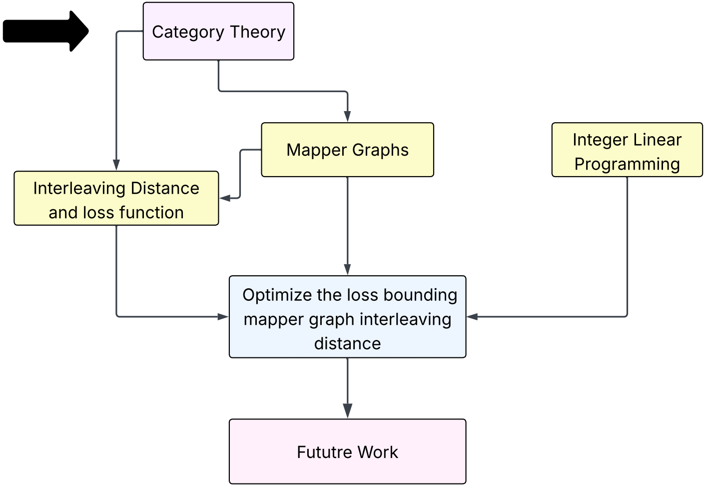
TL;DR

We make Mapper graphs ML-friendly
by
optimizing a loss that bounds interleaving distance.

Big Picture


Category Theory
Mapper Graphs
Integer Linear Programming
Interleaving Distance and loss function
Optimize the loss bounding mapper graph interleaving distance
Fututre Work

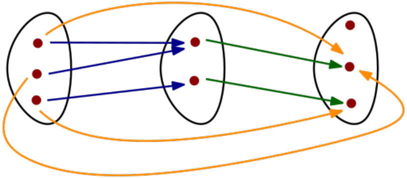
Big Picture

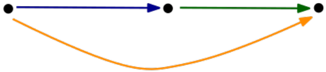


Category Theory
Mapper Graphs
Integer Linear Programming
Interleaving Distance and loss function
Optimize the loss bounding mapper graph interleaving distance
Fututre Work

Category Theory: Basics

- Study of structure across different areas.
- Relation between structures.
- Two main ingredients:
  - Objects
  - Morphisms
- Rules: composition + identity



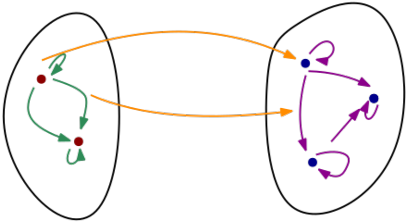


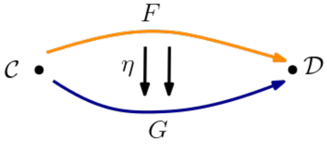
Set: category of sets

Category Theory: Maps Between Categories

- Functors: maps between categories
  - Objects $\rightarrow$ objects
  - Morphism $\rightarrow$ morphisms
- Natural transformations: Relates functors
- Natural transformations satisfy:

$$
\begin{array}{ccc}
X & \quad & F(X) \xrightarrow{\eta_X} G(X) \\
\downarrow f & & F(f)\downarrow \qquad\qquad \downarrow G(f) \\
Y & & F(Y) \xrightarrow{\eta_Y} G(Y)
\end{array}
$$





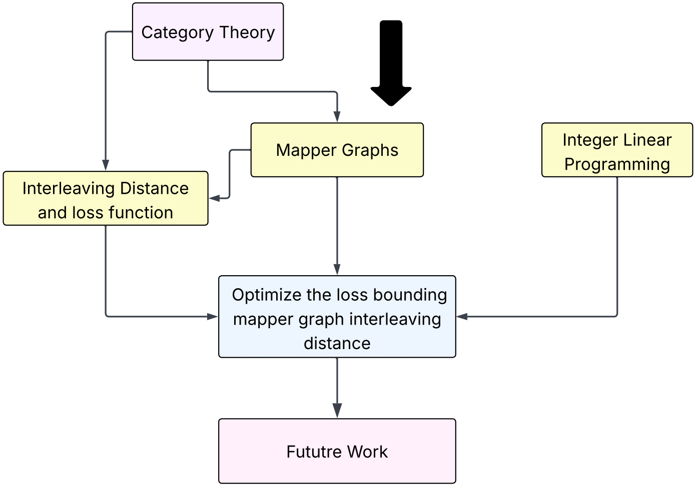
Functors and natural Transformations

Category Theory: Special Attention

$$F : \mathbf{Open}(\mathcal{U}) \rightarrow \mathbf{Set}$$

- $\mathbf{Open}(\mathcal{U})$: Category of open sets over $\mathcal{U}$
  - Objects: Open subsets of $\mathcal{U}$
  - Morphisms: Inclusions
- **Set**: Category of sets
  - Objects: Sets
  - Morphism: Set maps
- F maps
  - open sets to sets
  - inclusions to extension set maps
  - Satisfies consistency across overlaps $\leftarrow$ *cosheaf*

Big Picture



Category Theory
Mapper Graphs
Integer Linear Programming
Interleaving Distance and loss function
Optimize the loss bounding mapper graph interleaving distance
Fututre Work

Mapper Graphs

- A topological tool to summarize high-dimensional data
- Built using:
  - A lens function (real-valued)
  - A cover on the lens output
  - Clustering within each cover set
- Nodes $\rightarrow$ clusters, Edges $\rightarrow$ shared data points
- Captures shape and structure of data.

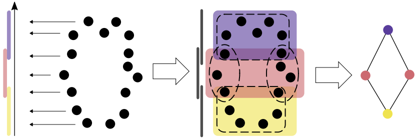


Image credit: Percival et al., 2024

Mapper Graphs: Categorical Framework

Goal: Express mappers as cosheaves.

- Given:
  - Data $\mathbb{X}$ with function $f : \mathbb{X} \to \mathbb{R}$
  - Cover $\mathcal{U} = \{U_\alpha\}$ of $\mathbb{R}$
- Preserves connected components of $f^{-1}(U_\alpha)$
- Discretize using a grid on $\mathbb{R}$
- Represent f in cosheaf form
- Encoded as functor: F : $\textbf{Open}(\mathcal{U}) \to \textbf{Set}$
- Components stored in $\pi_0(f^{-1}(U))$.

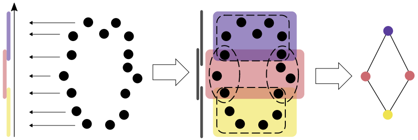
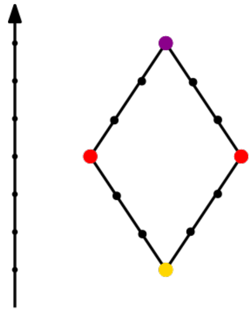


Image credit: Percival et al., 2024

Mapper Graphs: Categorical Framework

Goal: Express mappers as cosheaves.

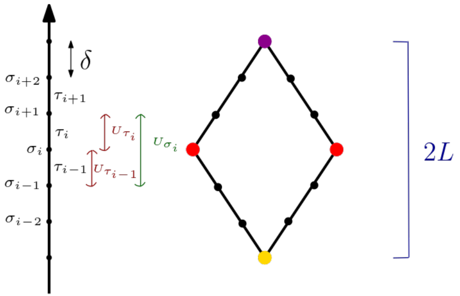
- Given:
  - Data $\mathbb{X}$ with function $f : \mathbb{X} \to \mathbb{R}$
  - Cover $\mathcal{U} = \{U_\alpha\}$ of $\mathbb{R}$
- Preserves connected components of $f^{-1}(U_\alpha)$
- Discretize using a grid on $\mathbb{R}$
- Represent f in cosheaf form
- Encoded as functor: $F : \textbf{Open}(\mathcal{U}) \to \textbf{Set}$
- Components stored in $\pi_0(f^{-1}(U))$.



Mapper Graphs: Categorical Framework

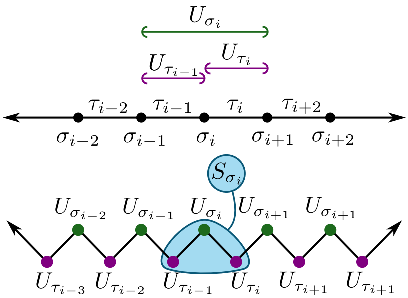
- Mapper bounded in $[-L, L]$.
- Cubical complex K of $\mathbb{R}$ with diameter δ.
- Cover $\mathcal{U} = \{U_{\sigma_i}\} \cup \{U_\tau\}$.
    - $U_{\sigma_i} = ((i-1)\delta, (i+1)\delta)$
    - $U_{\tau_i} = (i\delta, (i+1)\delta)$
- Open sets in $S \subseteq \mathcal{U}$ using Alexandrov topology.
    - Basis: $S_{\sigma_i} = \{U_{\tau_i}, U_{\sigma_i}, U_{\tau_{i+1}}\}$ and $S_{\tau_i} = \{U_{\tau_i}\}$
    - Geometrically, $|S_\sigma| = U_\sigma$ and $|S_\tau| = U_\tau$.
- Mapper $(\mathbb{X}, f)$ is given by

$$\begin{array}{rccc} F: & \textbf{Open}(\mathcal{U}) & \to & \textbf{Set} \\ & S & \mapsto & \pi_0(f^{-1}(|S|)) \end{array}$$

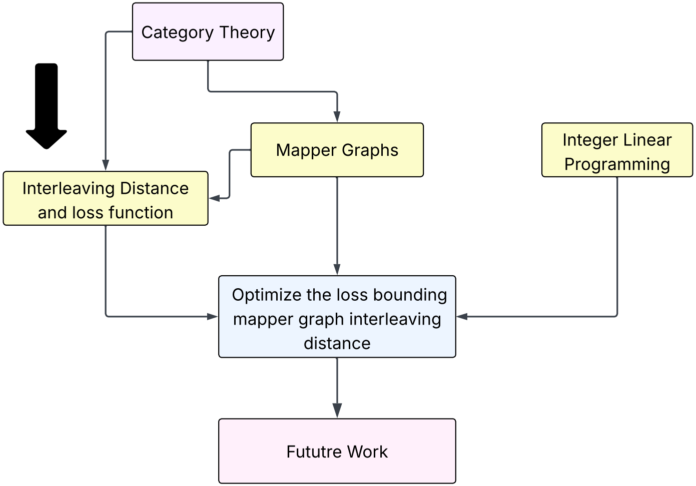


Mapper Graphs: Open Sets

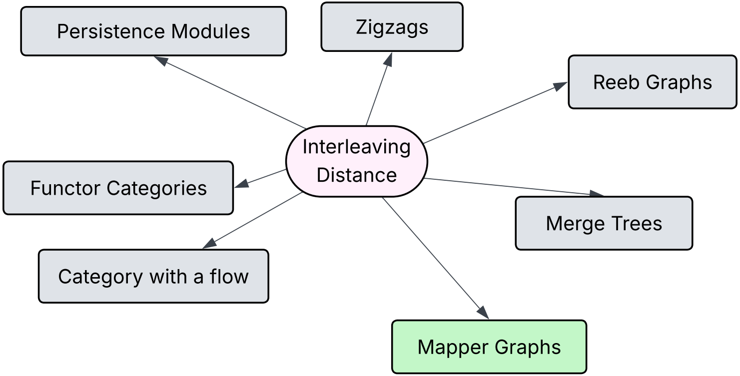
- Cover elements: U_{σ_i}, U_τ
- Open sets in Alexandrov topology: S_{σ_i}, S_{τ_i}
- Different, but same geometrically.



Big Picture


Category Theory
Mapper Graphs
Integer Linear Programming
Interleaving Distance and loss function
Optimize the loss bounding mapper graph interleaving distance
Fututre Work

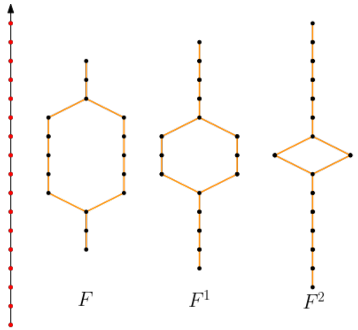
Interleaving Distance in TDA



Distance Between Mappers: Interleaving Distance

Goal: to compare $F, G : \mathbf{Open}(\mathcal{U}) \to \mathbf{Set}$.

- Define n-thickening of open sets.
  - Geometrically, $(i\delta, j\delta) \to ((i-n)\delta, ((j+n)\delta)$
- Thickening of functor $F^n := F \circ (-)^n$
  - Means, $F^n(S) = F(S^n)$
- n-interleaving is $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$.
- Must satisfy diagram commutativity.



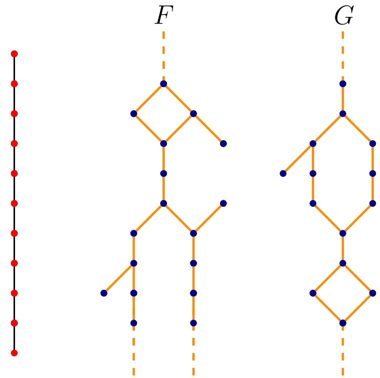
Distance Between Mappers: Interleaving Distance

- Diagrams to commute:

$$F(S) \xrightarrow{F[S\subseteq S^{2n}]} F(S^{2n}), \quad F(S) \xrightarrow{\varphi_S} G(S^n) \xrightarrow{\psi_{S^n}} F(S^{2n})$$

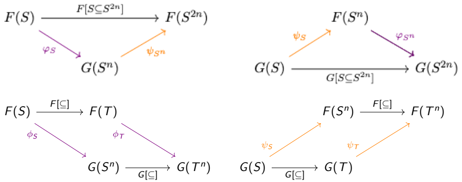
$$G(S) \xrightarrow{G[S\subseteq S^{2n}]} G(S^{2n}), \quad G(S) \xrightarrow{\psi_S} F(S^n) \xrightarrow{\varphi_{S^n}} G(S^{2n})$$

- Smallest n is the interleaving distance.
- NP hard.

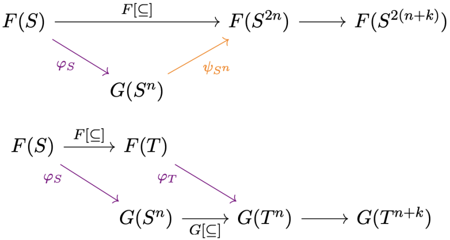


Bounding Interleaving Distance: Assignment

- Assignment (φ, ψ): Natural transformation like maps without commutativity.
- Find n to commute:
  - Triangle: For interleaving
  - Parallelogram: For natural transformation



- Given n, how much to thicken (k), so that the diagrams commute for $n+k$?



- Diagram loss $L_{\square}, L_{\square}, L_{\triangle}, L_{\triangledown}$ is a quality measure.

Bounding Interleaving Distance: Loss Function

- Loss $L(\varphi, \psi)$: Measures how far from commuting.

$$L_B(\phi, \psi) = \max_{\substack{\sigma < \tau \in K \\ \rho \in K}} \left\{ L_{\bigtriangledown}^{S_\tau, S_\sigma}, L_{\varpi}^{S_\tau, S_\sigma}, L_{\bigtriangledown}^{S_\rho, S_\rho^n}, L_{\varpi}^{S_\rho, S_\rho^n}, L_{\triangle}^{S_\rho}, L_{\triangledown}^{S_\rho} \right\}$$

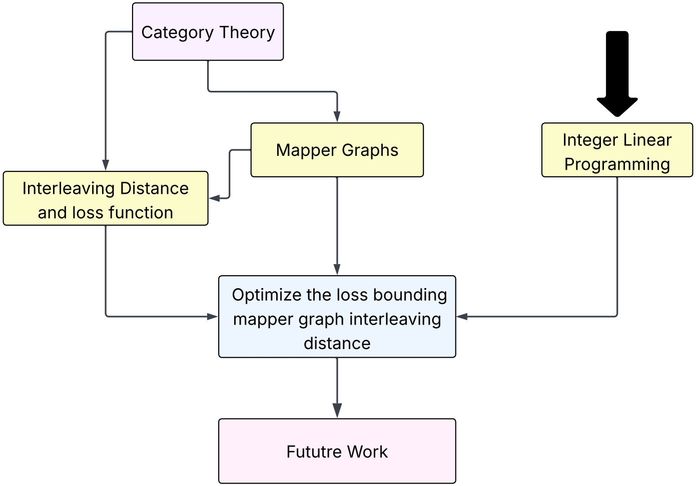
- Only on basis elements $L_B(\varphi, \psi)$.
- Computation is polynomial.

Theorem (Chambers et al. (2023))

For an n-assignment, $\varphi : F \Rightarrow G^n$ and $\psi : G \Rightarrow F^n$,

$$d_I(F, G) \leq n + L_B(\varphi, \psi).$$

Big Picture


Category Theory
Mapper Graphs
Integer Linear Programming
Interleaving Distance and loss function
Optimize the loss bounding mapper graph interleaving distance
Fututre Work

Integer Linear Programming

- Linear Programming (LP): optimize a linear function under linear constraints.
  - Objective function
  - Constraints
- Integer Linear Programming (ILP): same as LP, but variables must be integers.
- Used in scheduling, logistics, networks, etc.
- ILPs are harder to solve than LPs (NP-hard).
- General formulation:

$$
\begin{aligned}
\text{Maximize} \quad & \mathbf{c}^T \mathbf{x} \\
\text{Subject to} \quad & A\mathbf{x} \leq \mathbf{b} \\
& x_j \geq 0 \quad \forall j \in \{1, \cdots, n\}
\end{aligned}
$$

Integer Linear Programming

- Efficient implementation exists
- Python Library: PuLP
- Solver: CBC, Gurobi, CPLEX

Goal: Use ILP to optimize the loss

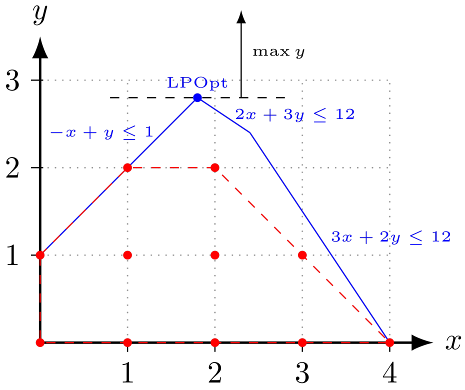
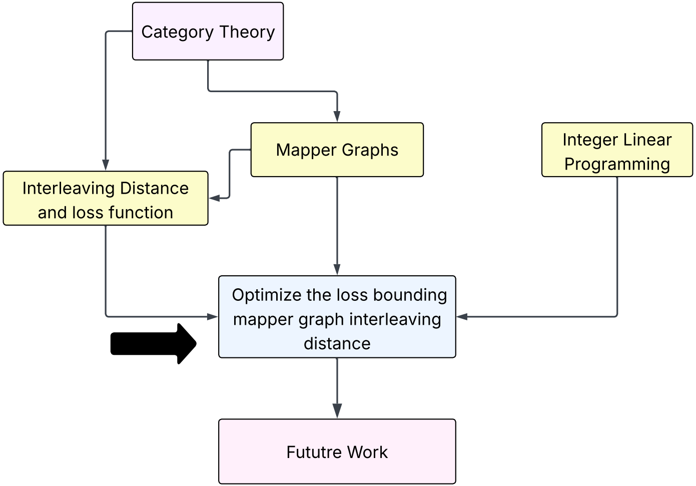


Image credits: Wikimedia commons

Big Picture



Category Theory
Mapper Graphs
Integer Linear Programming
Interleaving Distance and loss function
Optimize the loss bounding mapper graph interleaving distance
Fututre Work

Objective

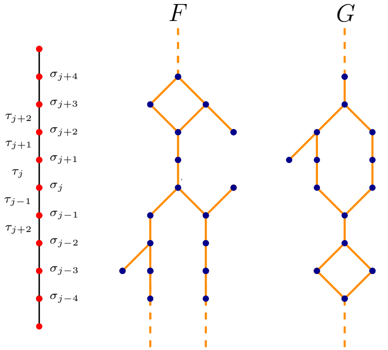
Can we optimize the loss function to get a better bound on the interleaving distance between two mapper graphs?

Preliminaries:

- Start with two mappers $F, G : \mathbf{Open}(\mathcal{U}) \to \mathbf{Set}$.
- Both mappers are in $B = [-L, L]$.
- Vertex at every integer function value.
- Only one connected component each.

Data Structure: Mapper as Graphs

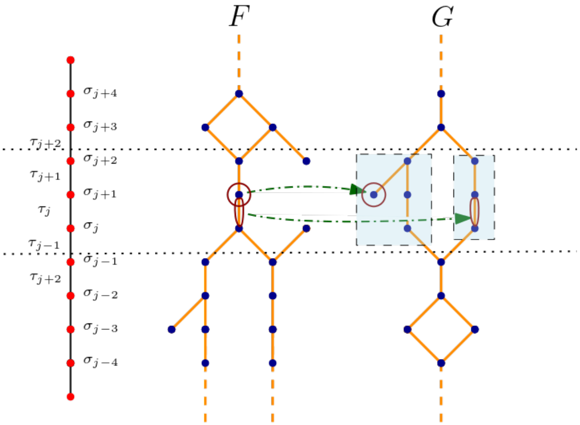
- Store F : $\mathbf{Open}(\mathcal{U}) \to \mathbf{Set}$ as graph $F \sim (V_F, E_F)$.
- Use grid structure on $\mathbb{R}$.
- Generate vertex for $F(S_{\sigma_i})$ and edge for $F(S_{\tau_i})$.
- $V_F = \coprod_{i=1}^{B} F(S_{\sigma_i})$ and $E_F = \coprod_{i=1}^{B-1} F(S_{\tau_i})$.
- Vertices: stored with height.
- *Assignment*: vertex and edge maps.
- Diagram commutes $\sim$ same connected component.



Commutativity and Connected Components

$$\begin{array}{ccccc} F(S_\tau) & \xrightarrow{F[\subseteq]} & F(S_\sigma) & & \\ & \searrow^{\varphi_{S_\tau}} & & \searrow^{\varphi_{S_\sigma}} & \\ & & G^n(S_\tau) & \xrightarrow[G[\subseteq]]{} & G^n(S_\sigma) \end{array}$$

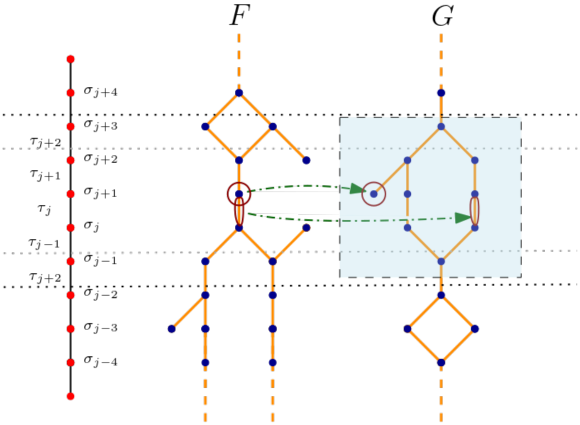
$$\begin{array}{ccccc} e & \longmapsto & v & & \\ & \searrow & & \searrow & \\ & & [e'] & \longmapsto & \begin{matrix}[w]\\ [e']\end{matrix} \end{array}$$



Commutative and Connected Components

$$
\begin{array}{ccccc}
F(S_\tau) & \xrightarrow{F[\subseteq]} & F(S_\sigma) & & \\
 & \searrow^{\varphi_{S_\tau}} & & \searrow^{\varphi_{S_\sigma}} & \\
 & & G^n(S_\tau) & \xrightarrow[G[\subseteq]]{} & G^n(S_\sigma) \longrightarrow G(S^{n+k}_{\sigma_\ell})
\end{array}
$$

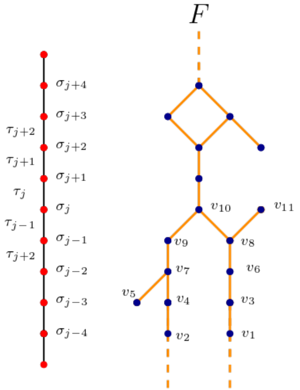
$$
\begin{array}{ccccc}
e & \longmapsto & v & & \\
 & \searrow & & \searrow & \\
 & & [e'] & \longmapsto & \begin{matrix}[w]\\ [e']\end{matrix} \longmapsto \begin{matrix}[w]\\ [e']\end{matrix}
\end{array}
$$



Data Structure: Maps as Matrices

Goal: Express diagram commutativity as matrix multiplication.

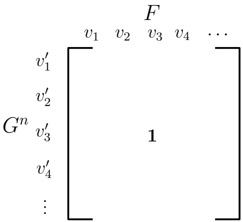
- Order vertices by increasing function values.
- Same for edges (lower vertex).
- Assignment: vertex and edge maps.
- Matrix whose rows and columns are these.
- Block Structure.

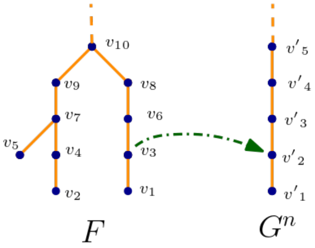


$$\{\sigma_{j-4} : (v_1, v_2), \sigma_{j-3} : (v_3, v_4), \cdots\}$$

Assignment Matrices

- Store maps like $\varphi : F \to G^n$.
- Place 1 if $\varphi(v) = v'$, 0 otherwise.
- For a valid map: only one 1 for each column.

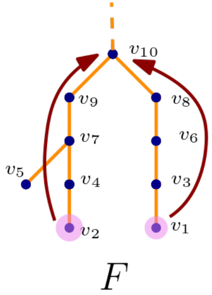




- To have lower loss, need better assignment.

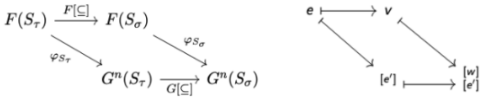
Other Matrices

- Distance Matrix:
  - How close are same-height vertices or edges to being connected?
  - Entries are positive integers.
- Boundary matrix: edge-vertex connection.
  - Up and down.
  - Binary
- Inclusion matrix: for $F \Rightarrow F^n$.
  - Binary.



Loss Terms as Matrix Multiplication

- Diagram:

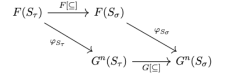
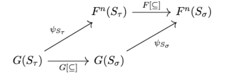
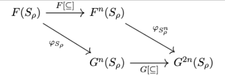
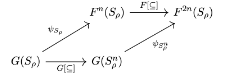
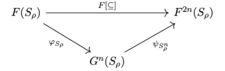
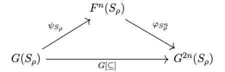


- Top-right: $M_\phi^V \cdot B_F^\downarrow$
- Left-Down: $B_{G^n}^\downarrow \cdot M_\phi^E$
- In matrix terms:

$$\max_i L_{\square}^{S_{\tau_i}, S_{\sigma_i}} = \max \left\{ x \mid x \in D_{G^n}^V \left(M_\phi^V \cdot B_F^\downarrow - B_{G^n}^\downarrow \cdot M_\phi^E \right) \right\}$$

- computing loss $\rightarrow$ finding largest matrix element.

Loss in Matrix Terms

	Loss Term	Diagram	Matrix Multiplication	Eval.
Edge-vertex Parallelogram	$L^{S_\tau,S_\sigma}_{\text{▱}}$		$D^V_{G^n}\left(M^V_\varphi \cdot B^\uparrow_F - B^\uparrow_{G^n} \cdot M^E_\varphi\right)$ $D^V_{G^n}\left(M^V_\varphi \cdot B^\downarrow_F - B^\downarrow_{G^n} \cdot M^E_\varphi\right)$	$\max\limits_{x_{ij}\in A} x_{ij}$
	$L^{S_\tau,S_\sigma}_{\text{▱}}$		$D^V_{F^n}\left(M^V_\psi \cdot B^\uparrow_G - B^\uparrow_{F^n} \cdot M^E_\psi\right)$ $D^V_{F^n}\left(M^V_\psi \cdot B^\downarrow_G - B^\downarrow_{F^n} \cdot M^E_\psi\right)$	
Thickening Parallelogram	$L^{S_\rho,S^n_\rho}_{\text{▱}}$		$D^V_{G^n}\left(M^V_{\varphi^n} \cdot I^V_F - I^V_{G^n} \cdot M^V_\varphi\right)$ $D^E_{G^n}\left(M^E_{\varphi^n} \cdot I^E_F - I^E_{G^n} \cdot M^E_\varphi\right)$	
	$L^{S_\rho,S^n_\rho}_{\text{▱}}$		$D^V_{F^n}\left(M^V_{\psi^n} \cdot I^V_G - I^V_{F^n} \cdot M^V_\psi\right)$ $D^E_{F^n}\left(M^E_{\psi^n} \cdot I^E_G - I^E_{F^n} \cdot M^E_\psi\right)$	
Triangle	$L^{S_\rho}_{\triangledown}$		$D^V_{F^{2n}}\left(I^V_{F^n} \cdot I^V_F - M^V_{\psi^n} \cdot M^V_\varphi\right)$ $D^E_{F^{2n}}\left(I^E_{F^n} \cdot I^E_F - M^E_{\psi^n} \cdot M^E_\varphi\right)$	$\max\limits_{x_{ij}\in A} \left\lceil \frac{x_{ij}}{2} \right\rceil$
	$L^{S_\rho}_{\triangle}$		$D^V_{G^{2n}}\left(I^V_{E^n} \cdot I^V_G - M^V_{\varphi^n} \cdot M^V_\psi\right)$ $D^E_{G^{2n}}\left(I^E_{G^n} \cdot I^E_G - M^E_{\varphi^n} \cdot M^E_\psi\right)$	

Implement Loss Optimization

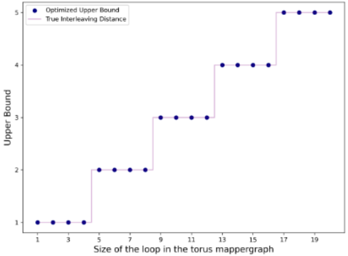
- Question: Can we formulate loss optimization as a linear program?
  - Yes! Both the objective function and constraints can be linearized.
- Discretized setup calls for integer linear programming.
- ILP is formulated as follows:

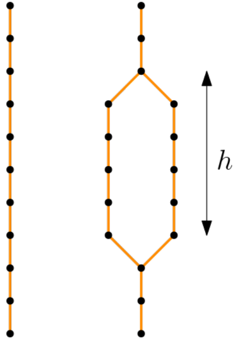
$$
\begin{aligned}
\text{Minimize} \quad & \ell \\
\text{Subject to} \quad & \ell \geq x_{ij}^{F,\uparrow} \quad \forall\, x_{ij}^{F,\uparrow} \in D_{G^n}^{V} \left(M_{\varphi}^{V} \cdot B_F^{\uparrow} - B_{G^n}^{\uparrow} \cdot M_{\varphi}^{E} \right) \\
& \sum_i x_{ij}^{\eta,A} = 1 \quad \forall x_{ij}^{\eta,A} \in M_{\eta}^{A},\ \eta \in \{\phi, \phi^n, \psi, \psi^n\},\ A \in \{V, E\} \\
& x_{ij} \geq 0 \\
& \vdots \quad \text{(more constraints)}
\end{aligned}
$$

- Nonlinearity in triangles (e.g., $M_{\psi^n}^{V} \cdot M_{\varphi}^{V}$) is linearized with additional variables.

Experiments with Small Mappers

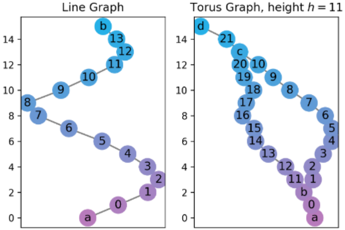
- Small mappers where interleaving distance can be computed.
- Line mapper: One vertex at every height.
- Torus mapper: A loop in the middle.
- Interleaving distance is $\lceil \frac{h}{4} \rceil$.

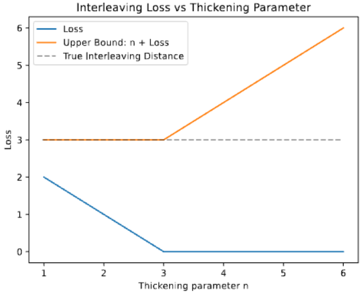




Experiments with Small Mappers

- Interleaving can be achieved for $n = 1$

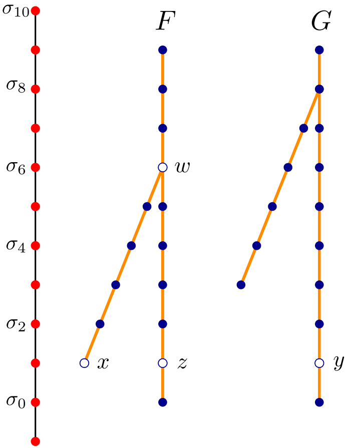


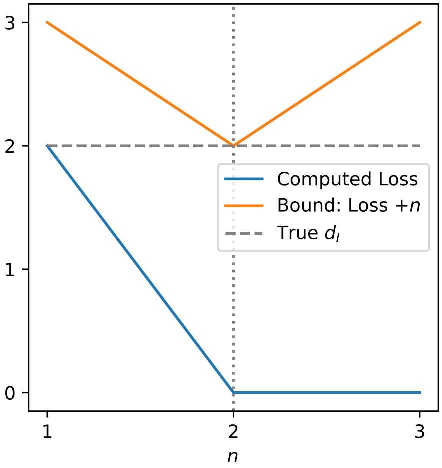


- Interleaving: $h = 11$, $\lceil \frac{h}{4} \rceil = 3$.

Optimization with $n = 1$ is not Enough

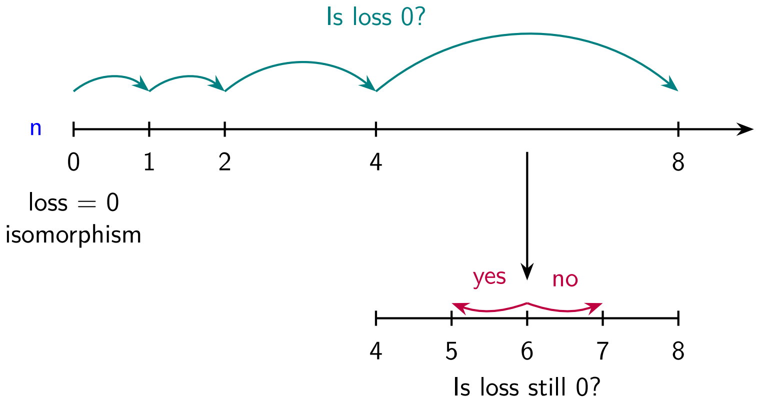
- Found mappers where upper bound goes down for higher n.





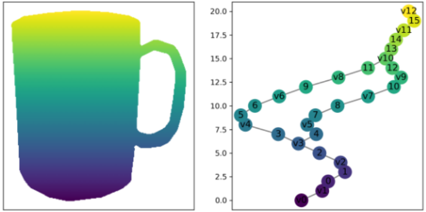
Search for Optimum n

- $n = 1$ does not always give best upper bound ($n + loss$).
- Combination of binary and exponential search for $n \in 0, 1, \cdots, 2L$.



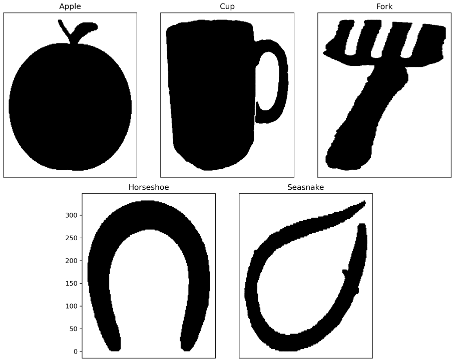
Experiments with Images

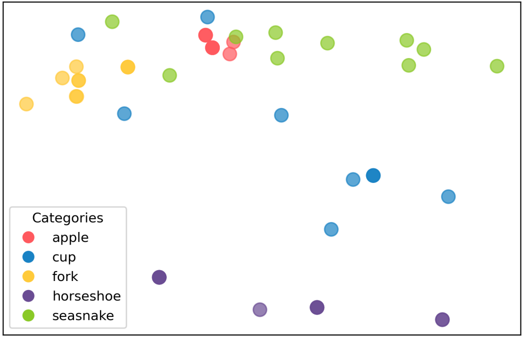
- MPEG7 image dataset.
- Compute mappers on different objects.
- Y-coordinates as lens function.
- Compare using optimized loss.



Experiments with Images

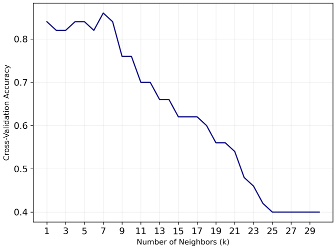
- Pairwise distance with the optimized mapper loss
- MDS to preserve pairwise dissimilarities

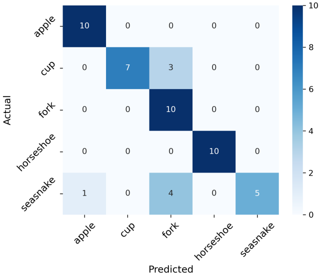




Experiments with Images

- KNN classification with K=7.
- Accuracy achieved 84%.
- Caveat: Highly dependent on the mapper parameters.





Pre-print Available Now

Cornell University

We gratefully acknowledge support from the Simons Foundation, Michigan State University, and all contributors. Donate

arXiv > cs > arXiv:2504.03865

Search... All fields Search
Help | Advanced Search

Computer Science > Computational Geometry

[Submitted on 4 Apr 2025]

Towards an Optimal Bound for the Interleaving Distance on Mapper Graphs

Erin Wolf Chambers, Ishika Ghosh, Elizabeth Munch, Sarah Percival, Bei Wang

Mapper graphs are a widely used tool in topological data analysis and visualization. They can be viewed as discrete approximations of Reeb graphs, offering insight into the shape and connectivity of complex data. Given a high-dimensional point cloud $\mathbb{X}$ equipped with a function $f : \mathbb{X} \to \mathbb{R}$, a mapper graph provides a summary of the topological structure of $\mathbb{X}$ induced by f, where each node represents a local neighborhood, and edges connect nodes whose corresponding neighborhoods overlap. Our focus is the interleaving distance for mapper graphs, arising from a discretization of the version for Reeb graphs, which is NP-hard to compute. This distance quantifies the similarity between two mapper graphs by measuring the extent to which they must be ``stretched" to become comparable. Recent work introduced a loss function that provides an upper bound on the interleaving distance for mapper graphs, which evaluates how far a given assignment is from being a true interleaving. Finding the loss is computationally tractable, offering a practical way to estimate the distance.
In this paper, we employ a categorical formulation of mapper graphs and develop the first framework for computing the associated loss function. Since the quality of the bound depends on the chosen assignment, we optimize this loss function by formulating the problem of finding the best assignment as an integer linear programming problem. To evaluate the effectiveness of our optimization, we apply it to small mapper graphs where the interleaving distance is known, demonstrating that the optimized upper bound successfully matches the interleaving distance in these cases. Additionally, we conduct an experiment on the MPEG-7 dataset, computing the pairwise optimal loss on a collection of mapper graphs derived from images and leveraging the distance bound for image classification.

Subjects: **Computational Geometry (cs.CG)**
MSC classes: 55N31
Cite as: arXiv:2504.03865 **[cs.CG]**
(or arXiv:2504.03865v1 **[cs.CG]** for this version)
https://doi.org/10.48550/arXiv.2504.03865

Submission history
From: Ishika Ghosh [view email]
[v1] Fri, 4 Apr 2025 18:43:01 UTC (1,279 KB)

Access Paper:
View PDF
TeX Source
Other Formats
view license

Current browse context:
cs.CG
< prev | next >
new | recent | 2025-04
Change to browse by:
cs

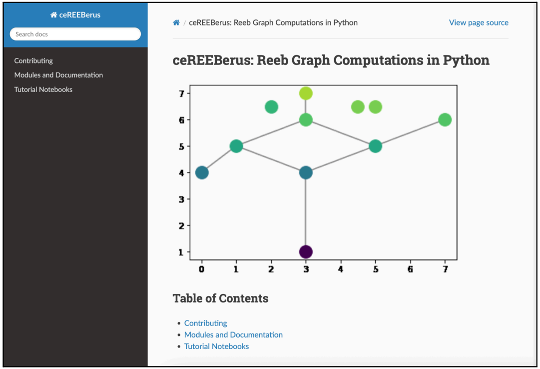
References & Citations
NASA ADS
Google Scholar
Semantic Scholar

Export BibTeX Citation

Bookmark

Open-Source Shoutout: ceREEBerus!

ceReeberus is a Python package for working with Reeb graphs, in particular with a view towards using the interleaving distance in an applied context.



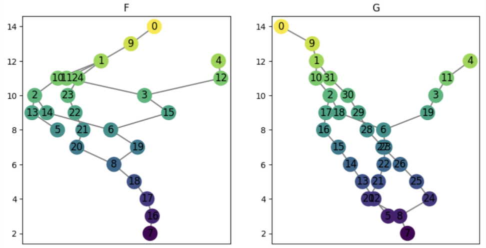
Find it here:

Tutorial: Compute Optimized Upper Bound on Interleaving

- Two mapper graphs, `F` and `G`.
- Initialize
  - `interleave = Interleave(F,G)`
- Optimize the loss
  - interleave.fit()
- Obtain optimized upper bound
  - interleave.n

[2]:
```
# Start with two example MapperGraphs
F = ex_mg.interleave_example_A()
G = ex_mg.interleave_example_B()

# Plot the two MapperGraphs
fig, ax = plt.subplots(1, 2, figsize=(10, 5))
F.draw(ax = ax[0])
ax[0].set_title('F')
G.draw(ax = ax[1])
ax[1].set_title('G');
```



Passing in the two mapper graphs and then fitting will return the best found interleaving value.

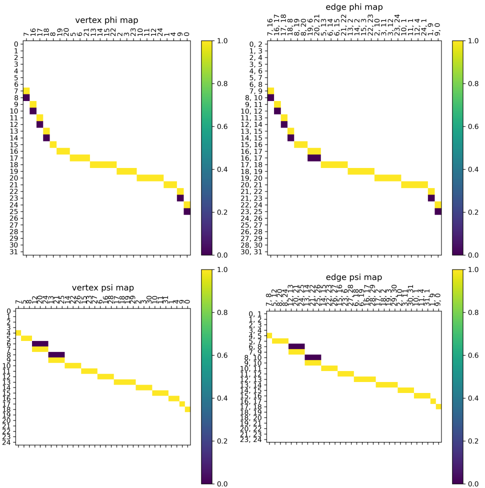
[3]:
```
# Interleave the two MapperGraphs
myInt = Interleave(F, G)
myInt.fit()

print('The found interleaving bound is d_I(F,G) <=', myInt.n)
```

```
The found interleaving bound is d_I(F,G) <= 4
```

Tutorial: Extracting Optimized Interleaving Maps

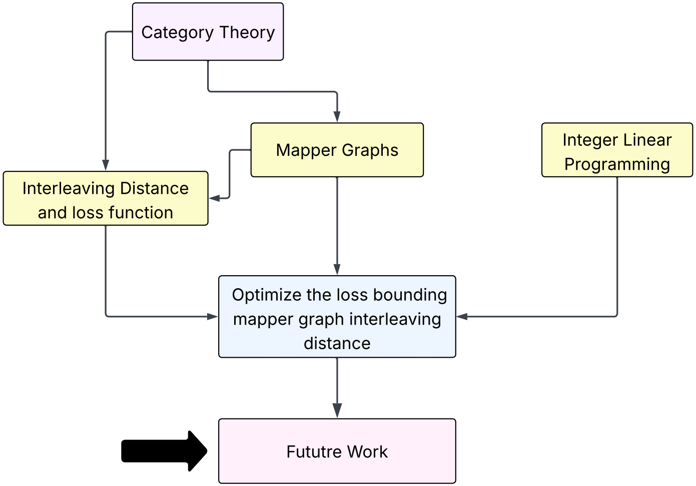
- Optimized interleaving maps `phi`, `psi`
- Stored as labeled block diagonal matrices
- Visualize the maps
  - `interleave.phi(obj_type='V').draw()`
  - Use `obj_type='E'` for edge maps



To Summarize

- First available method to compute a bound for the interleaving distance.
- Code for understanding interleavings on mapper graphs.
- Mapper as graphs, maps as matrices.
- Loss computation as matrix multiplication.
- Optimize using integer linear programming.
- Experiments to establish the idea.

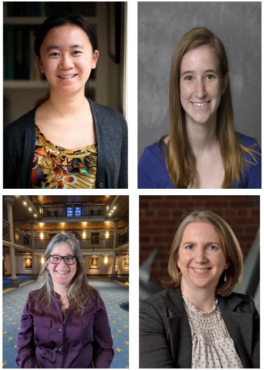
Big Picture


Category Theory
Mapper Graphs
Integer Linear Programming
Interleaving Distance and loss function
Optimize the loss bounding mapper graph interleaving distance
Fututre Work

Future Work

- Have more experiments with different datasets.
- Improve efficiency of loss optimization.
- Compare with similar methods.
- Focus on how mappers are being generated.
- Can we evaluate mapper parameter selection?
- Use in ML pipelines.

Thank You!



Coauthors

MunchLab, Spring 2025

email: ghoshis3@msu.edu

Find me here:

Chambers, E. W., Munch, E., Percival, S., and Wang, B. (2023). Bounding the interleaving distance for geometric graphs with a loss function. *arXiv preprint arXiv:2307.15130*.