Advances in Graph Persistence

Alessandro Mella

Dip. di Matematica, Univ. di Bologna, Italia

04 June 2019

Based on work by: M.G. Bergomi, M. Ferri, A. Tavaglione, L. Zuffi

Outline

1. **Introduction**
2. **Topological graph persistence**
3. **Combinatorial graph persistence**
   - An abstract setting
   - Coherent samplings
   - Steady and ranging sets
4. **An application: Hubs**
5. **Conclusions**

All our software concerning graph persistence can be found at

`https://gitlab.com/mattia.bergomi/perscomb`

1 **Introduction**

2 Topological graph persistence

3 Combinatorial graph persistence
- An abstract setting
- Coherent samplings
- Steady and ranging sets

4 An application: Hubs

5 Conclusions

More and more often, (weighted) graphs represent data and data structures:

- images as graphs of regions
- 3D shapes as graphs of volumes
- power grids
- neural networks
- communication networks
- social networks
- ...

On the other hand, persistent topology offers a modular, powerful tool for analyzing data of various nature.

On the other hand, persistent topology offers a modular, powerful tool for analyzing data of various nature.

After a brief recall of the use of persistent homology on simplicial complexes obtained from weighted graphs, we present a general definition of *persistence functions* which allows the use of persistence diagrams without going through simplicial or topological constructions.

On the other hand, persistent topology offers a modular, powerful tool for analyzing data of various nature.

After a brief recall of the use of persistent homology on simplicial complexes obtained from weighted graphs, we present a general definition of *persistence functions* which allows the use of persistence diagrams without going through simplicial or topological constructions.

Two general ways of producing persistence functions are presented, with examples: *Coherent samplings* and *steady* and *ranging sets*.

1 Introduction

2 **Topological graph persistence**

3 Combinatorial graph persistence
- An abstract setting
- Coherent samplings
- Steady and ranging sets

4 An application: Hubs

5 Conclusions

Graphs as complexes

Graphs will be simple throughout, and will be thought of as 1D simplicial complexes. Homomorphisms, and in particular isomorphisms, will be simplicial maps.

Graphs as complexes

Graphs will be simple throughout, and will be thought of as 1D simplicial complexes. Homomorphisms, and in particular isomorphisms, will be simplicial maps.

Given a weighted graph (G, f), where $G = (V, E)$ and $f : E \to \mathbb{R}$ is a *filtering function*, one can extend f to a filtering function $\bar{f} : V \cup E \to \mathbb{R} \cup \{\infty\}$ by defining it as ∞ on isolated vertices and on any other vertex v as the minimum value of f on its incident edges.

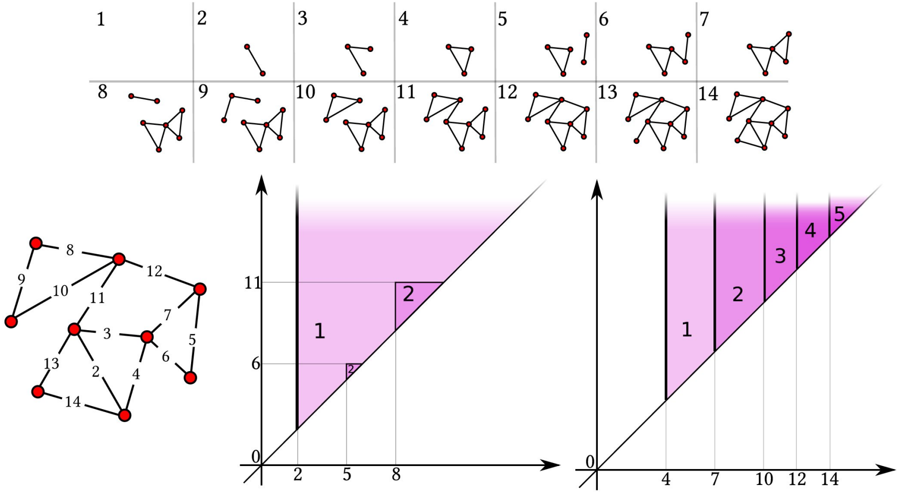
Graphs as complexes

Graphs will be simple throughout, and will be thought of as 1D simplicial complexes. Homomorphisms, and in particular isomorphisms, will be simplicial maps.

Given a weighted graph (G, f), where $G = (V, E)$ and $f : E \to \mathbb{R}$ is a *filtering function*, one can extend f to a filtering function $\bar{f} : V \cup E \to \mathbb{R} \cup \{\infty\}$ by defining it as ∞ on isolated vertices and on any other vertex v as the minimum value of f on its incident edges.

Therefore, a weighted graph gives rise to persistent Betti number functions and persistence diagrams in a natural way.

Graphs as complexes



Complexes from graphs

It is well-known that it's possible to build simplicial complexes out of a graph G in various ways. Simplices can be its

Complexes from graphs

It is well-known that it's possible to build simplicial complexes out of a graph G in various ways. Simplices can be its

- cliques
- neighbourhoods
- enclaveless sets

and many others.

Complexes from graphs

It is well-known that it's possible to build simplicial complexes out of a graph G in various ways. Simplices can be its

- cliques
- neighbourhoods
- enclaveless sets

and many others.

Cliques and neighbourhoods (and clique communities; see later) have been considered in weighted graphs (G, f) for using persistent homology in several applied settings.

Complexes from graphs

D. Horak, S. Maletić, and M. Rajković. *Persistent homology of complex networks*. Journal of Statistical Mechanics: Theory and Experiment, 2009(03):P03034, 2009.

W. Huang and A. Ribeiro. *Persistent homology lower bounds on high-order network distances*. IEEE Transactions on Signal Processing, 65(2):319–334, 2017.

S. Pal, T. J. Moore, R. Ramanathan, and A. Swami. *Comparative topological signatures of growing collaboration networks*. In Workshop on Complex Networks CompleNet, pages 201–209. Springer, 2017.

G. Petri, P. Expert, F. Turkheimer, R. Carhart-Harris, D. Nutt, P. J. Hellyer, and F. Vaccarino. *Homological scaffolds of brain functional networks*. Journal of The Royal Society Interface, 11(101):20140873, 2014.

M. W. Reimann, M. Nolte, M. Scolamiero, K. Turner, R. Perin, G. Chindemi, P. Dłotko, R. Levi, K. Hess, and H. Markram. *Cliques of neurons bound into cavities provide a missing link between structure and function*. Frontiers in Computational Neuroscience, 11:48, 2017.

D.S. Bassett, and O. Sporns. *Network neuroscience*. Nature neuroscience, 20.3: 353, 2017.

1 Introduction

2 Topological graph persistence

3 Combinatorial graph persistence
- An abstract setting
- Coherent samplings
- Steady and ranging sets

4 An application: Hubs

5 Conclusions

Is topology necessary for persistence?

In topological graph persistence, we sought hereditary structures, so that we could build simplicial complexes.

Can we do without this constraint? Can we still have persistence diagrams and all the machinery we are used to?

Is topology necessary for persistence?

In topological graph persistence, we sought hereditary structures, so that we could build simplicial complexes.

Can we do without this constraint? Can we still have persistence diagrams and all the machinery we are used to?

We shall show that this is indeed possible, by defining *persistence functions* in an abstract way.

General Setting

Consider a concrete category $(\mathcal{C}, \mathcal{U})$, with $\mathcal{U} : \mathcal{C} \to$ **Sets** a faithful functor. For $X \in \mathcal{C}$, define the category $\mathcal{C}_X$ of subobjects of X.

Definition 3.1

We say that $(\mathcal{C}, \mathcal{U})$ has canonical subobjects if

1. $\mathcal{C}$ has pullbacks and the functor $\mathcal{U}$ preserves pullbacks
2. for every object $X \in \mathcal{C}$ and for every $Z \subseteq \mathcal{U}(X)$, if there is a subobject $T \overset{\chi}{\hookrightarrow} X$ such that $Z = \mathcal{U}(\chi)(\mathcal{U}(T))$, then the category $\mathcal{C}_{X|Z}$ has a terminal object $U \hookrightarrow X$
3. every morphism $Y \overset{\chi}{\to} X$ can be factored as $Y \overset{\phi}{\to} W \overset{\psi}{\hookrightarrow} X$, where ψ is a monomorphism and $\mathcal{U}(\psi)(\mathcal{U}(W)) = \mathcal{U}(\chi)(\mathcal{U}(Y))$

Filtrations

Definition 3.2

Let $\mathcal{R}$ be the poset category of real numbers. We define a filtration in $\mathcal{C}$ to be a functor $\mathcal{F} : \mathcal{R} \to \mathcal{C}$ such that if $u < v$ then $\mathcal{F}(u)$ is a subobject of $\mathcal{F}(v)$.

Proposition 3.3

Let $X \in$ Obj($\mathcal{C}$) and $f : \mathcal{U}(X) \to \mathbb{R}$ be an inferiorly bound function such that for any $t \in \mathbb{R}$ there is at least one subobject $X_t \xhookrightarrow{\chi_t} X$ with $\mathcal{U}(X_t) = f^{-1}((-\infty, t])$. Let $Y_t \xhookrightarrow{v_t} X$ be the canonical subobject associated to $f^{-1}((-\infty, t])$. Then $\mathcal{F}_{(X,f)}$ defined by $\mathcal{F}_{(X,f)}(t) = Y_t$ is a filtration in $\mathcal{C}$ and $\mathcal{S}_{(X,f)} = \mathcal{U} \circ \mathcal{F}$ is a filtration in $\mathcal{U}(\mathcal{C})$.

Graph setting

In the following $\mathcal{C}$ will be the category **Graph**, and for each graph G, the category of subobjects will contain its subgraphs.

Filtrations of subobjects will clearly be filtrations of subgraphs.

Natural pseudodistance

Let now $(G, f), (G', f')$, with $G = (V, E), G' = (V', E')$ be weighted graphs and H be the (possibly empty) set of isomorphisms from G to G'.

Definition 3.4

The *natural* pseudodistance of (G, f) and (G', f') is

$$\delta\big((G, f), (G', f')\big) = \begin{cases} \infty & \text{if} \quad H = \emptyset \\ \inf_{\phi \in H} \sup_{e \in E} |f(e) - f'\big(\phi(e)\big)| & \text{otherwise} \end{cases}$$

Persistence functions

We set $\Delta^+ = \{(u, v) \in \mathbb{R} \mid u < v\}$, $\Delta = \{(u, v) \in \mathbb{R} \mid u = v\}$ and $\overline{\Delta}^+ = \Delta^+ \cup \Delta$.

Let (G, f) be any weighted graph. For each $t \in \mathbb{R}$, the *sublevel graph* G_t is the subgraph of G induced by $f^{-1}\big((-\infty, t]\big)$.

Assume we have a function Λ_G defined on all inclusions between subgraphs of G, with values in the nonnegative integers, and such that $\Lambda_G(\iota) = 0$ if ι has the empty set as domain. Define $\lambda_{(G,f)}(u, v) = \Lambda_G(\iota)$, where ι is the inclusion of G_u into G_v.

Persistence functions

Definition 3.5

All functions $\lambda_{(G,f)} : \Delta^+ \to \mathbb{Z}$ are said to be *persistence functions* if conditions 1 and 2 are satisfied; they are said to be *stable* persistence functions if also 3 holds:

1. $\lambda_{(G,f)}(u, v)$ is nondecreasing in u and nonincreasing in v;
2. for all $u_1, u_2, v_1, v_2 \in \mathbb{R}$ such that $u_1 \leq u_2 < v_1 \leq v_2$ the following inequality holds:
   $\lambda_{(G,f)}(u_2, v_1) - \lambda_{(G,f)}(u_1, v_1) \geq \lambda_{(G,f)}(u_2, v_2) - \lambda_{(G,f)}(u_1, v_2)$
3. given an analogous pair (G', f'), if an isomorphism $\psi : G \to G'$ exists such that $\sup_{e \in E} |f(e) - f'(\psi(e))| \leq h$ $(h > 0)$, then for all $(u, v) \in \Delta^+$ the inequality $\lambda_{(G,f)}(u - h, v + h) \leq \lambda_{(G',f')}(u, v)$ holds.

Stability

Remark 3.6

A set of theorems holds, granting that any persistence function (conditions 1 and 2) $\lambda_{(G,f)}$ has the same structure as Persistent Betti Numbers functions. In particular, it can be summarized by a *persistence diagram* $D(f)$ with the usual cornerpoints (proper and at infinity).
$d\big(D(f), D(f')\big)$ will be the usual bottleneck distance.

Stability

Remark 3.6

A set of theorems holds, granting that any persistence function (conditions 1 and 2) $\lambda_{(G,f)}$ has the same structure as Persistent Betti Numbers functions. In particular, it can be summarized by a *persistence diagram* $D(f)$ with the usual cornerpoints (proper and at infinity).
$d\big(D(f), D(f')\big)$ will be the usual bottleneck distance.

Theorem 3.7 (Stability)

For weighted graphs $(G, f), (G', f')$ *as above, if* $\lambda_{(G,f)}$ *and* $\lambda_{(G',f')}$ *are stable then*

$$d\big(D(f), D(f')\big) \le \delta\big((G, f), (G', f')\big)$$

Stability

Related with stability, we have the problem of universality: Is the inequality of Thm. 3.7 the best one that we can obtain from persistence diagrams?

Coherent samplings

A first way of building persistence functions is the following.

Definition 3.8

A *coherent sampling* $\mathcal{V}$ is the assignment to each graph G, where $G = (V, E)$ of a set $\mathcal{V}(G)$ of subsets of $V \cup E$, such that the following conditions 1 and 2 hold; it will be said to be a *stable* coherent sampling if also condition 3 holds:

1. each $\mathcal{V}(G)$ is finite (possibly empty);
2. if G is a subgraph of H, then each element of $\mathcal{V}(G)$ is contained in exactly one element of $\mathcal{V}(H)$;
3. if $\psi : G \to G'$ is an-isomorphism, then $\mathcal{V}(G') = \psi\big(\mathcal{V}(G)\big)$.

For each inclusion $\iota : G \to H$ let $\Lambda(\iota)$ be the number of elements of $\mathcal{V}(H)$ containing at least one element of $\mathcal{V}(G)$.

Coherent samplings

Proposition 3.9

Let a coherent sampling $\mathcal{V}$ be given; for all graphs $G = (V, E)$, for all filtering functions $f : E \to \mathbb{R}$, let $\lambda_{(G,f)} : \Delta^+ \to \mathbb{Z}$ be defined by $\lambda_{(G,f)}(u, v) = \Lambda(\iota_{(u,v)})$ where $\iota_{(u,v)} : G_u \to G_v$ is the inclusion homomorphism.
Then the functions $\lambda_{(G,f)}$ are persistence functions. If the coherent sampling is stable, so are the persistence functions.

Example 1: Blocks

We recall that in a (loopless) graph G a *cut vertex* (or *separating vertex*) is a vertex $v \in V(G)$ whose deletion (along with incident edges) makes the number of connected components of G increase. A *block* is a connected graph which does not contain any cut vertex. A block of a graph G is a maximal subgraph H such that H is a block.

Example 1: Blocks

We recall that in a (loopless) graph G a *cut vertex* (or *separating vertex*) is a vertex $v \in V(G)$ whose deletion (along with incident edges) makes the number of connected components of G increase. A *block* is a connected graph which does not contain any cut vertex. A block of a graph G is a maximal subgraph H such that H is a block.

Proposition 3.10

The assignment $\mathcal{B}$, which maps each graph G to the set of its blocks, is a stable coherent sampling.

Example 1: Blocks

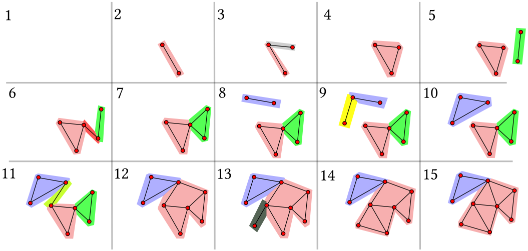
Definition 3.11

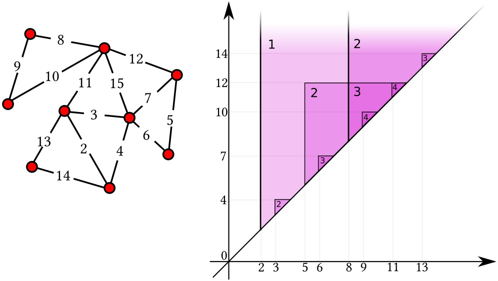
Given a weighted graph (G, f), we call *persistent block number* the function $bl_{(G,f)} : \Delta^+ \to \mathbb{Z}$ which maps the pair (u, v) to the number of blocks of G_v containing at least one block of G_u.

Corollary 3.12

$bl_{(G,f)}$ is a stable persistence function.

Example 1: Blocks





Example 1: Blocks

Theorem 3.13 (Universality)

If $\tilde{d}$ is a distance for persistent block diagrams such that

$$\tilde{d}\big(D_{bl}(f), D_{bl}(f')\big) \leq \delta\big((G, f), (G', f')\big)$$

for any persistent block diagrams $D_{bl}(f)$, $D_{bl}(f')$ of weighted graphs (G, f), (G', f'), with G, G' isomorphic, then

$$\tilde{d}\big(D_{bl}(f), D_{bl}(f')\big) \leq d\big(D_{bl}(f), D_{bl}(f')\big)$$

Example 2: Edge-blocks

We recall that in a graph G a *cut edge* (or *bridge*) is an edge $e \in E(G)$ whose deletion makes the number of connected components of G increase. We define an *edge-block* as a connected graph which contains at least one edge, but does not contain any cut edge. An edge-block of a graph G is a maximal subgraph H such that H is an edge-block.

Example 2: Edge-blocks

We recall that in a graph G a *cut edge* (or *bridge*) is an edge $e \in E(G)$ whose deletion makes the number of connected components of G increase. We define an *edge-block* as a connected graph which contains at least one edge, but does not contain any cut edge. An edge-block of a graph G is a maximal subgraph H such that H is an edge-block.

Proposition 3.14

The assignment $\mathcal{E}$, which maps each graph G to the set of its edge-blocks, is a stable coherent sampling. □

Example 2: Edge-blocks

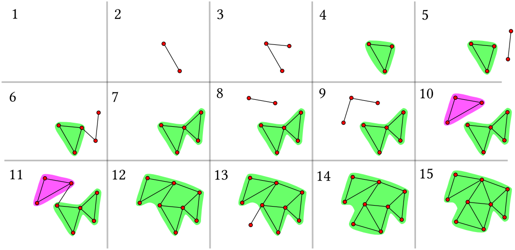
We recall that in a graph G a *cut edge* (or *bridge*) is an edge $e \in E(G)$ whose deletion makes the number of connected components of G increase. We define an *edge-block* as a connected graph which contains at least one edge, but does not contain any cut edge. An edge-block of a graph G is a maximal subgraph H such that H is an edge-block.

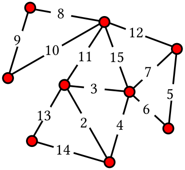
Proposition 3.14

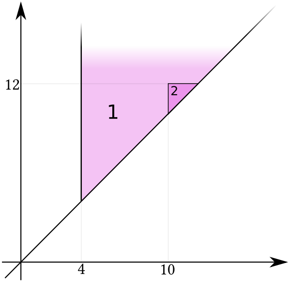
The assignment $\mathcal{E}$, which maps each graph G to the set of its edge-blocks, is a stable coherent sampling. □

The definition of a *persistent edge-block number* function $ebl_{(G,f)}$, its stability and universality also hold.

Example 2: Edge-blocks







Example 3: Clique communities

Given a graph $G = (V, E)$, two of its k-cliques (i.e. cliques of k vertices) are said to be *adjacent* if they share $k - 1$ vertices; a k*−clique community* is a maximal union of k-cliques such that any two of them are connected by a sequence of k-cliques, where each k-clique of the sequence is adjacent to the following one. This construction has been applied to network analysis and also to weighted graphs.

Example 3: Clique communities

Given a graph $G = (V, E)$, two of its k-cliques (i.e. cliques of k vertices) are said to be *adjacent* if they share $k - 1$ vertices; a $k-$*clique community* is a maximal union of k-cliques such that any two of them are connected by a sequence of k-cliques, where each k-clique of the sequence is adjacent to the following one. This construction has been applied to network analysis and also to weighted graphs.

Proposition 3.15

The assignment $\mathcal{C}^k$, which maps each graph G to the set of its k-clique communities, is a stable coherent sampling.

Example 3: Clique communities

Given a graph $G = (V, E)$, two of its k-cliques (i.e. cliques of k vertices) are said to be *adjacent* if they share $k - 1$ vertices; a $k-$*clique community* is a maximal union of k-cliques such that any two of them are connected by a sequence of k-cliques, where each k-clique of the sequence is adjacent to the following one. This construction has been applied to network analysis and also to weighted graphs.

Proposition 3.15

The assignment $\mathcal{C}^k$, which maps each graph G to the set of its k-clique communities, is a stable coherent sampling.

The definition of a *persistent k-clique community number* function $cc^k_{(G,f)}$, its stability and universality also hold.

Example 3: Clique communities

S. Fortunato. *Community detection in graphs*. Physics reports, 486(3-5):75–174, 2010.

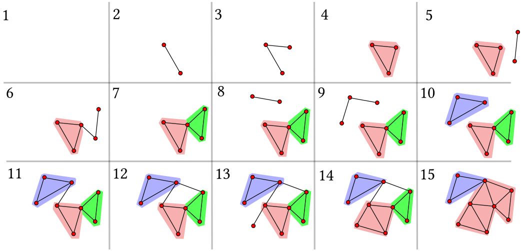
J. M. Kumpula, J.-P. Onnela, J. Saramäki, K. Kaski, and J. Kertész. *Emergence of communities in weighted networks*. Physical review letters, 99(22):228701, 2007.

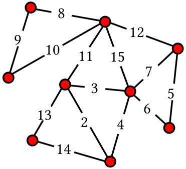
G. Palla, A.-L. Barabási, and T. Vicsek. *Quantifying social group evolution*. Nature, 446(7136):664, 2007.

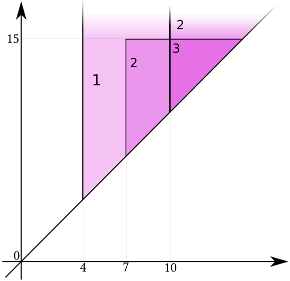
B. Rieck, U. Fugacci, J. Lukasczyk, and H. Leitte. *Clique community persistence: A topological visual analysis approach for complex networks*. IEEE transactions on visualization and computer graphics, 24(1):822–831, 2018.

R. Toivonen, J.-P. Onnela, J. Saramäki, J. Hyvönen, and K. Kaski. *A model for social networks*. Physica A: Statistical Mechanics and its Applications, 371(2):851–860, 2006.

Example 3: Clique communities







Steady and ranging

Given a graph $G = (V, E)$, let $F : 2^{V \cup E} \to \{true, false\}$ be any feature. We call *F-set* any set $A \subseteq V \cup E$ such that $F(A) = true$.

Let now the weighted graph (G, f) be given. Given any real number w, we shall say that $A \subseteq V \cup E$ is an F-set *at level w* if it is an F-set of the subgraph G_w.

Steady and ranging

Given a graph $G = (V, E)$, let $F : 2^{V \cup E} \to \{true, false\}$ be any feature. We call *F-set* any set $A \subseteq V \cup E$ such that $F(A) = true$.

Let now the weighted graph (G, f) be given. Given any real number w, we shall say that $A \subseteq V \cup E$ is an F-set *at level w* if it is an F-set of the subgraph G_w.

Definition 3.16

We call $A \subseteq V \cup E$ a *steady* F-set (or simply an s-F-set) at (u, v) ($(u, v) \in \Delta^+$) if it is an F-set at all levels w with $u \leq w \leq v$. We call A a *ranging* F-set (or simply an r-F-set) at (u, v) if there exist levels $w \leq u$ and $w' \geq v$ at which it is an F-set.

Let $SF_{(G,f)}(u, v)$ be the set of s-F-sets at (u, v) and let $RF_{(G,f)}(u, v)$ be the set of r-F-sets at (u, v).

Steady and ranging

Proposition 3.17

The function which assigns to $(u, v) \in \Delta^+$ the number $|SF_{(X,f)}(u, v)|$ is a persistence function.

Proposition 3.18

The function which assigns to $(u, v) \in \Delta^+$ the number $|RF_{(X,f)}(u, v)|$ is a persistence function.

Example: Eulerian sets

Given any graph $G = (V, E)$, we define $Eu : 2^{V \cup E} \to \{true, false\}$ to yield *true* on a set A if and only if A is a set of vertices whose induced subgraph of G is nonempty, connected, Eulerian and maximal with respect to these properties; in that case A is said to be a *Eu*-set of G.

Let now (G, f) be a weighted graph. We apply Def. 3.16 to feature Eu for a weighted graph (G, f).

Example: Eulerian sets

Definition 3.19

Given any real number w, the set of vertices A is a *Eu*-set *at level* w if it is a *Eu*-set of the subgraph G_w.

It is a *steady Eu*-set (an s-*Eu*-set) at (u, v) ($(u, v) \in \Delta^+$) if it is a *Eu*-set at all levels w with $u \leq w \leq v$.

It is a *ranging Eu*-set (an r-*Eu*-set) at (u, v) if there exist levels $w \leq u$ and $w' \geq v$ at which it is a *Eu*-set.

$SEu_{(G,f)}(u, v)$ and $REu_{(G,f)}(u, v)$ are respectively the sets of s-*Eu*-sets and of r-*Eu*-sets at (u, v).

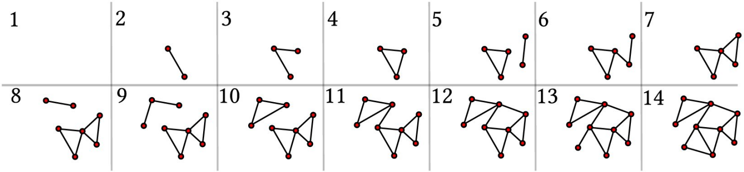
Example: Eulerian sets

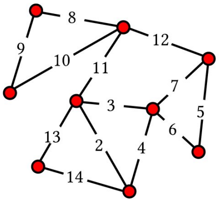
Proposition 3.20

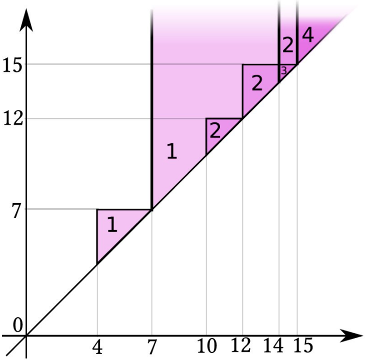
The function which assigns to $(u, v) \in \Delta^+$ *the number* $|SEu_{(G,f)}(u, v)|$ *and the function which assigns to* $(u, v) \in \Delta^+$ *the number* $|REu_{(G,f)}(u, v)|$ *are persistence functions.*

Both functions can be proved to be unstable.

Example: Eulerian sets







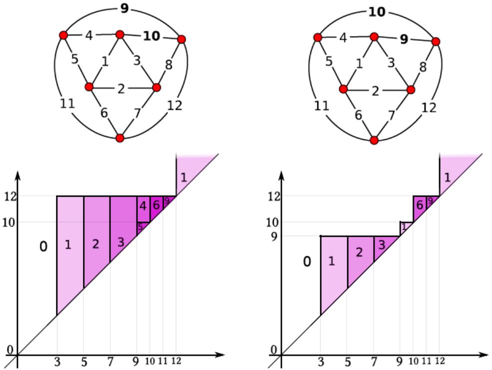
Stability

It is important to point out that stability is not always guaranteed. For the example shown, it has been proven that

1. Blocks
2. Edge-Blocks
3. Cliques community

are not only stable, but we also have that the universality property is satisfied. Instead a conterexample for the stability of Eulerian sets has been found.

Instability of Eulerian sets



1 Introduction

2 Topological graph persistence

3 Combinatorial graph persistence
- An abstract setting
- Coherent samplings
- Steady and ranging sets

4 **An application: Hubs**

5 Conclusions

Steady and ranging hubs

We now apply Props. 3.17 and 3.18 and a smart idea of V. Kurlin to the study of "hubs" in networks. As always, (G, f) is given, with $G = (V, E)$.

The property we are going to use gives *false* for all subsets of $V \cup E$ apart from the singletons formed by vertices whose degree is greater than or equal to the degree of all their neighbors:

Steady and ranging hubs

We now apply Props. 3.17 and 3.18 and a smart idea of V. Kurlin to the study of “hubs” in networks. As always, (G, f) is given, with $G = (V, E)$.

The property we are going to use gives *false* for all subsets of $V \cup E$ apart from the singletons formed by vertices whose degree is greater than or equal to the degree of all their neighbors:

Definition 4.1

A *temporary hub (t-hub) at level u* is a vertex of G_u whose degree is greater than or equal to the degree of its neighbors.

Steady and ranging hubs

Definition 4.2

A *steady hub (s-hub) at* (u, v) ($(u, v) \in \Delta^+$) is a vertex which is a t-hub at all levels w with $u \leq w \leq v$.

Definition 4.3

A *ranging hub (r-hub) at* (u, v) ($(u, v) \in \Delta^+$) is a vertex such that there exist levels $w \leq u$ and $w' \geq v$ at which it is a t-hub.

Steady and ranging hubs

We define $\sigma_{(G,f)} : \Delta^+ \to \mathbb{Z}$ as follows: For every $(u, v) \in \Delta^+$, $\sigma_{(G,f)}(u, v)$ is the number of s-hubs at (u, v).

Proposition 4.4

σ is a persistence function.

We define $\varrho_{(G,f)} : \Delta^+ \to \mathbb{Z}$ as follows: For every $(u, v) \in \Delta^+$, $\varrho_{(G,f)}(u, v)$ is the number of r-hubs at (u, v).

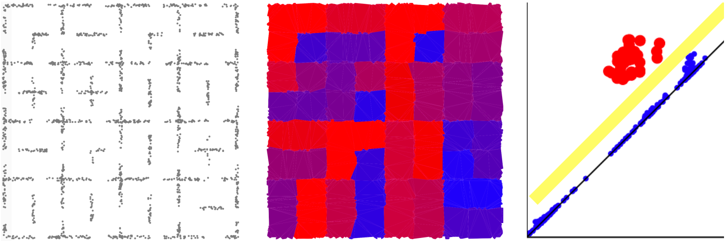
Proposition 4.5

ϱ is a persistence function.

Both functions can be proved to be unstable.

Hub selection

Finally, we use Kurlin's *widest diagonal gap* for selecting the top s- and r-hubs.



V. Kurlin. *A fast persistence-based segmentation of noisy 2D clouds with provable guarantees*. Pattern recognition letters, 83:3-12, 2016.

Example 1: US airports

A first application of the search for relevant hubs has been done on a set of major US airports plus two Canadian ones. The edges connect airports between which there are regular flights.

Example 1: US airports

A first application of the search for relevant hubs has been done on a set of major US airports plus two Canadian ones. The edges connect airports between which there are regular flights.

As filtering functions we use:

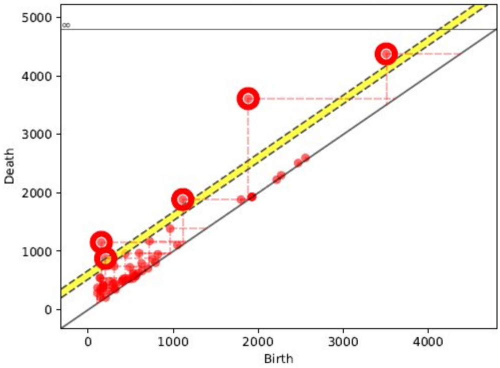
- distance
- weekly flight frequency
- their product

and their opposites (+their maximum)

Example 1: US airports

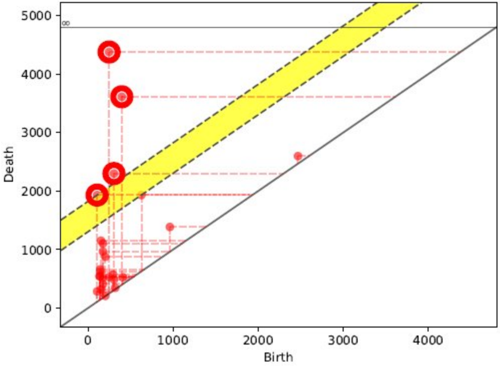
Albuquerque	Atlanta	Baltimore	Boston	Buffalo
Cheyenne	Chicago	Cincinnati	Cleveland	Dallas
Denver	Detroit	El Paso	Houston	Indianapolis
Jacksonville	Kansas City	Las Vegas	Los Angeles	Memphis
Miami	Milwaukee	Mobile	Montreal	New Orleans
New York	Oakland/Emeryville	Philadelphia	Phoenix	Pittsburgh
Portland	Sacramento	Salt Lake City	San Antonio	San Diego
San Francisco	Seattle	St. Louis	St. Paul-Minneapolis	Tampa
Toronto	Tucson	Vancouver	Washington	

Example 1: US airports (distance)



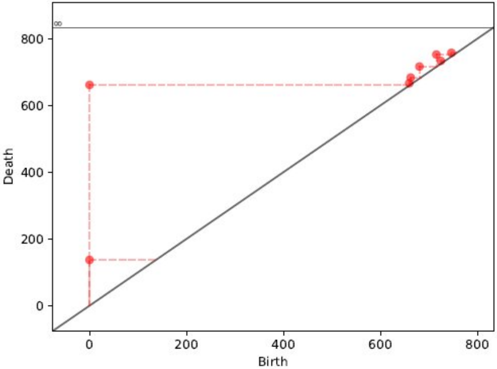
Steady hubs above 2nd widest gap							
5 cornerpoints, 4 vertices							
Atlanta	2	Dallas	1	Detroit	1	Seattle	1

Example 1: US airports (distance)



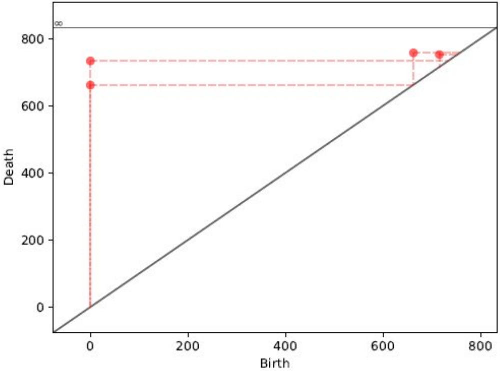
Ranging hubs above 3rd widest gap			
4 cornerpoints, 4 vertices			
Atlanta	Chicago	Dallas	Houston

Example 1: US airports (max - frequency)



Steady hubs							
8 cornerpoints, 4 hubs							
Atlanta	2	Chicago	4	Dallas	1	New York	1

Example 1: US airports (max - frequency)



Ranging hubs			
4 cornerpoints, 4 hubs			
Atlanta	Chicago	Dallas	New York

Example 2: European languages

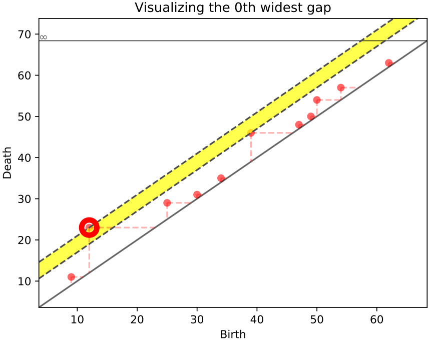
A second application is on languages of the European Union plus Turkish:

Castilian	Catalan	Croatian	Czech	Danish
Dutch	English	Finnish	French	Galitian
German	Greek	Hungarian	Italian	Polish
Portuguese	Romanian	Swedish	Turkish	

The graph is complete.

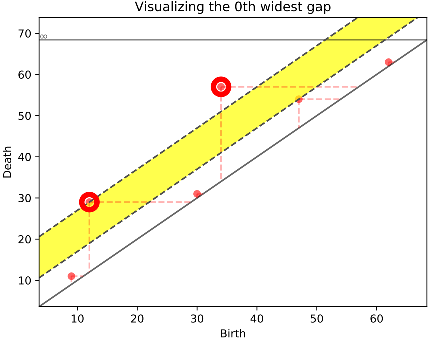
Filtering function is the opposite of the percentage of common properties (+ its max).

Example 2: European languages



Steady hubs					
11 cornerpoints, 6 vertices					
Castilian	1	Catalan	2	Dutch	1
English	2	Portuguese	4	Swedish	1

Example 2: European languages



Ranging hubs		
6 cornerpoints, 6 vertices		
Castilian English	Catalan Portuguese	Dutch Swedish

Set persistence: a non-graph example

Definition 4.6

Let $k \in \mathbb{N}$. We define the neighbor set of a pixel $(i, j) = x \in I$ of dimension k to be

$$N_k(x) = \{x' = (i', j') : i' = i + m, j' = j + n, -k \leq m, n \leq k\}$$

Definition 4.7

Let $m, n \in \mathbb{N}$ be such that $|N_k(x)| - n > m$ for any pixel $x \in I$. We say that a pixel $x \in I$ is active at level $l \in \mathbb{Z}$ if the following conditions are satisfied

1. $|N_k(x) \cap f^{-1}([-\infty, l])| \geq m$
2. $|N_k(x) \setminus f^{-1}([-\infty, l])| \leq n$

Set persistence: a non-graph example

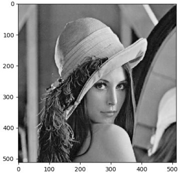
We define $\gamma_{(G,f)} : \Delta^+ \to \mathbb{Z}$ as follows: For every $(u, v) \in \Delta^+$, $\sigma_{(G,f)}(u, v)$ is the number of active pixels at (u, v).

Proposition 4.8

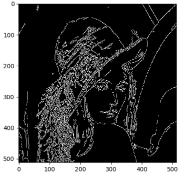
γ is a persistence function.

We applied this idea to the identification of borders of objects inside an image. We used a threshold to select the relevant cornerpoints (and thus pixels) and compared our results to Canny algorithm for edge detection.

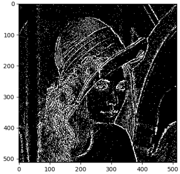
Set persistence: a non-graph example



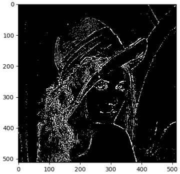
(a) Original



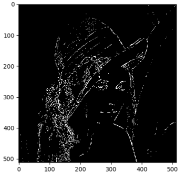
(b) Canny



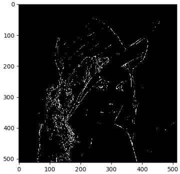
(c) Our: $t = 5$



(d) Our: $t = 10$



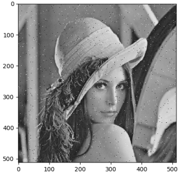
(e) Our: $t = 15$



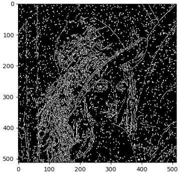
(f) Our: $t = 20$

Figure 1: The original image and the results of Canny and our algorithms

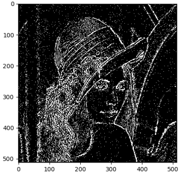
Set persistence: a non-graph example



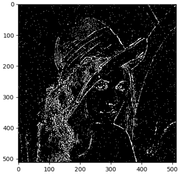
(a) Original



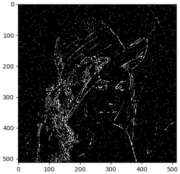
(b) Canny



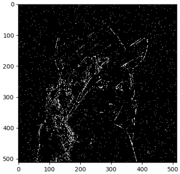
(c) Our: $t = 5$



(d) Our: $t = 10$



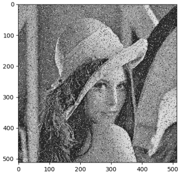
(e) Our: $t = 15$



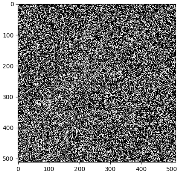
(f) Our: $t = 20$

Figure 2: The original image and the results of Canny and our algorithms

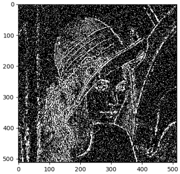
Set persistence: a non-graph example



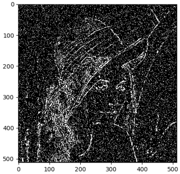
(a) Original



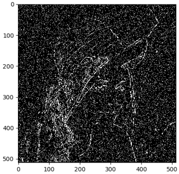
(b) Canny



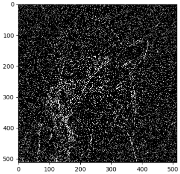
(c) Our: $t = 5$



(d) Our: $t = 10$



(e) Our: $t = 15$



(f) Our: $t = 20$

Figure 3: The original image and the results of Canny and our algorithms

1 Introduction

2 Topological graph persistence

3 Combinatorial graph persistence
- An abstract setting
- Coherent samplings
- Steady and ranging sets

4 An application: Hubs

5 **Conclusions**

Signatures of data by persistence diagrams are possible also without the need of topological constructions.

Signatures of data by persistence diagrams are possible also without the need of topological constructions.

This is possible in the domain of weighted graphs, but we are going to extend this possibility to a wider context.

Signatures of data by persistence diagrams are possible also without the need of topological constructions.

This is possible in the domain of weighted graphs, but we are going to extend this possibility to a wider context.

Here we have presented two general construction methods for persistence functions:

- Coeherent samplings
- Steady and ranging sets.

Furure Works

Some of the future development could be:

1. Multidimensional persistence
2. Extend to more general categorical setting
3. From hubs to clustering
4. Other aplication: e.g. Hough transform

THANKS FOR YOUR ATTENTION !

mella.alessandro3@unibo.it

All our software concerning graph persistence can be found at
`https://gitlab.com/mattia.bergomi/perscomb`