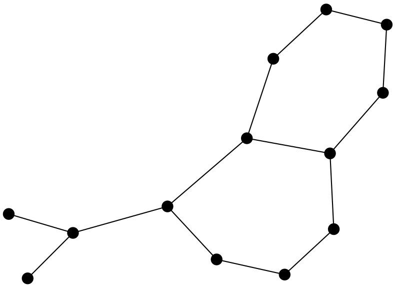
Curvature for Graph Learning

Bastian Rieck (@Pseudomanifold)

HELMHOLTZ MUNICH

Part I: Some Graph Learning Paradigms

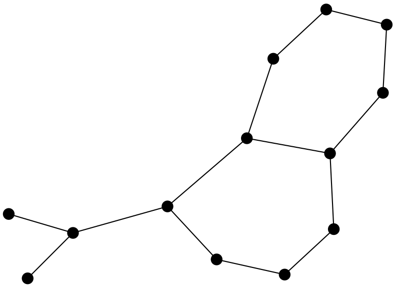
What is *graph learning*?



Typical Tasks

- Graph/node/edge classification

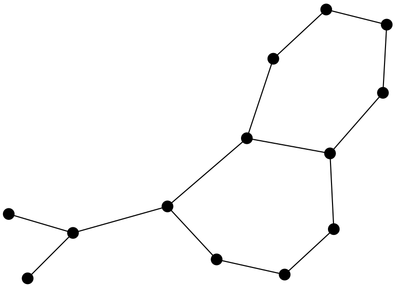
What is *graph learning*?



Typical Tasks

- Graph/node/edge classification
- Graph/node/edge regression

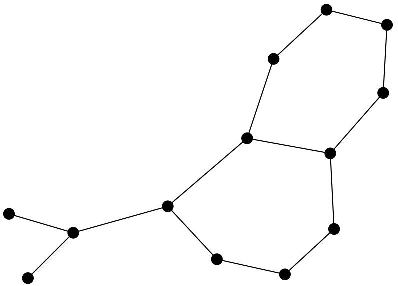
What is *graph learning*?



Typical Tasks

- Graph/node/edge classification
- Graph/node/edge regression
- Edge prediction

What is *graph learning*?



Typical Tasks

- Graph/node/edge classification
- Graph/node/edge regression
- Edge prediction
- Graph distribution comparison

How to *represent* graphs?

- Two graphs G and G' generally have a *different* number of vertices.

How to *represent* graphs?

- Two graphs G and G' generally have a *different* number of vertices.
- Hence, we require ways to *vectorise* graphs via a map f, such that $f(G), f(G') \in \mathbb{R}^d$.

How to *represent* graphs?

- Two graphs G and G' generally have a *different* number of vertices.
- Hence, we require ways to *vectorise* graphs via a map f, such that $f(G), f(G') \in \mathbb{R}^d$.
- The map f needs to be *permutation-invariant*, i.e. oblivious to the ordering of the graph.

What are typical *algorithms* for representing graphs?

Shallow approaches

- node2vec (encoder–decoder)
- Graph kernels (RKHS feature maps)
- Laplacian-based embeddings

Deep approaches

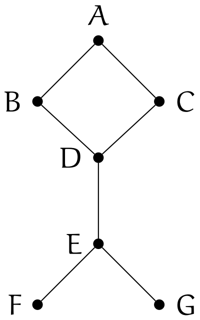
- Graph convolutional networks
- Graph isomorphism networks
- Graph attention networks

Message passing

The predominant paradigm in graph machine learning

Concept

Neighbouring nodes can exchange *messages* x, y, z (vectors in $\mathbb{R}^d$), which are *aggregated* (via a sum, a mean, or other permutation-invariant functions).

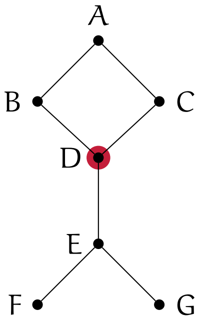


Message passing

The predominant paradigm in graph machine learning

Concept

Neighbouring nodes can exchange *messages* x, y, z (vectors in $\mathbb{R}^d$), which are *aggregated* (via a sum, a mean, or other permutation-invariant functions).

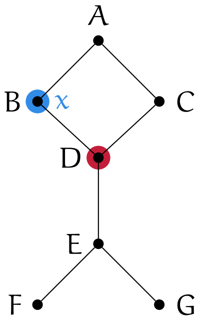


Message passing

The predominant paradigm in graph machine learning

Concept

Neighbouring nodes can exchange *messages* x, y, z (vectors in $\mathbb{R}^d$), which are *aggregated* (via a sum, a mean, or other permutation-invariant functions).

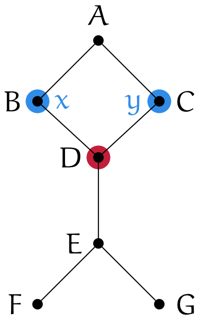


Message passing

The predominant paradigm in graph machine learning

Concept

Neighbouring nodes can exchange *messages* x, y, z (vectors in $\mathbb{R}^d$), which are *aggregated* (via a sum, a mean, or other permutation-invariant functions).

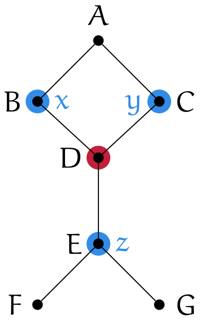


Message passing

The predominant paradigm in graph machine learning

Concept

Neighbouring nodes can exchange *messages* x, y, z (vectors in $\mathbb{R}^d$), which are *aggregated* (via a sum, a mean, or other permutation-invariant functions).

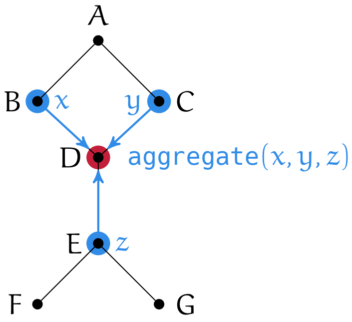


Message passing

The predominant paradigm in graph machine learning

Concept

Neighbouring nodes can exchange *messages* x, y, z (vectors in $\mathbb{R}^d$), which are *aggregated* (via a sum, a mean, or other permutation-invariant functions).

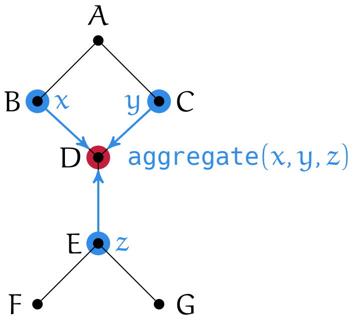


Message passing

The predominant paradigm in graph machine learning

Concept

Neighbouring nodes can exchange *messages* x, y, z (vectors in $\mathbb{R}^d$), which are *aggregated* (via a sum, a mean, or other permutation-invariant functions).



Moral

Informative *global* representations can arise from entirely *local* measurements.

Part II: A Brief Introduction to Curvature

What is curvature?

Motivation

Characterise how 'curved' an object (a surface, a manifold, a topological space, ...) is. Curvature can be *extrinsic* or *intrinsic*.

What is curvature?

Motivation

Characterise how 'curved' an object (a surface, a manifold, a topological space, …) is. Curvature can be *extrinsic* or *intrinsic*.

Gaussian curvature

Gaussian curvature K is the product of the *principal curvatures* κ_1, κ_2. It is an intrinsic property of a surface and does not depend on a specific embedding.

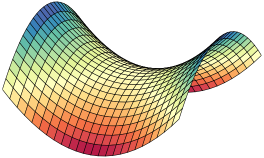
What is curvature?

Motivation

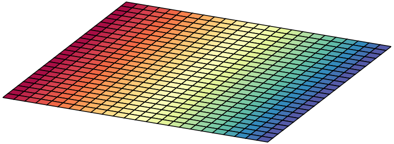
Characterise how 'curved' an object (a surface, a manifold, a topological space, ...) is. Curvature can be *extrinsic* or *intrinsic*.

Gaussian curvature

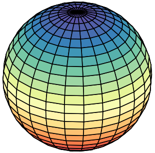
Gaussian curvature K is the product of the *principal curvatures* κ_1, κ_2. It is an intrinsic property of a surface and does not depend on a specific embedding.



$K < 0$



$K = 0$



$K > 0$

Geometrical characterisations of curvature

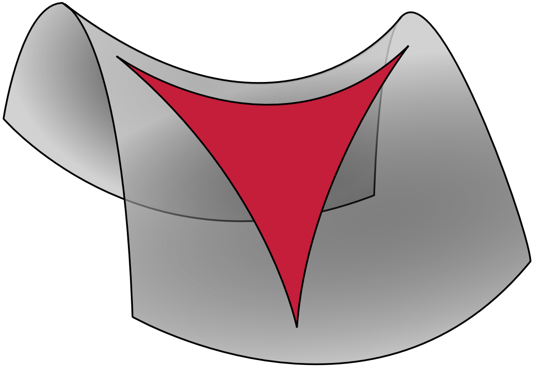
Negative curvature

In negative curvature, geodesic triangles are ‘thinner’ than reference triangles and exhibit ‘angular defects.’

Geometrical characterisations of curvature

Negative curvature

In negative curvature, geodesic triangles are 'thinner' than reference triangles and exhibit 'angular defects.'



Geometrical characterisations of curvature

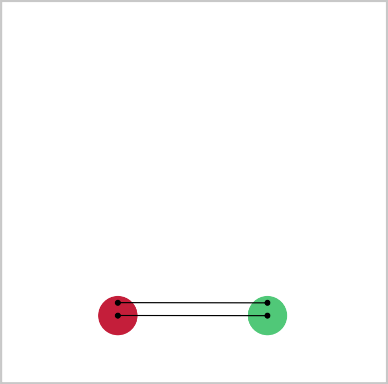
Positive curvature

In positive Ricci curvature, *corresponding* (using parallel transport) points of spheres are *closer than their respective centres are*.

Geometrical characterisations of curvature

Positive curvature

In positive Ricci curvature, *corresponding* (using parallel transport) points of spheres are *closer than their respective centres are*.

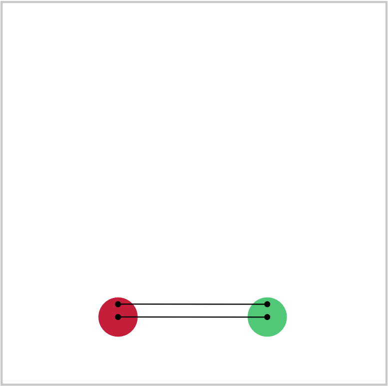


Flat

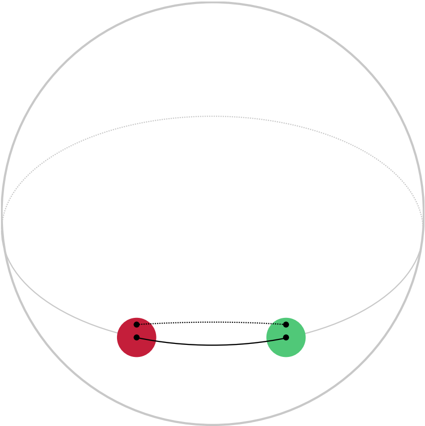
Geometrical characterisations of curvature

Positive curvature

In positive Ricci curvature, *corresponding* (using parallel transport) points of spheres are *closer than their respective centres are*.



Flat



Positively curved

Why is curvature useful?

Manifold setting

Curvature, despite being a *local* quantity, provides information about *global* characteristics of a manifold.

Gauss–Bonnet

$$\int_M K \, dA + \int_{\partial M} k_g \, ds = 2\pi\chi(M)$$

Bonnet–Myers

Let $\mathcal{M}$ be a d-dimensional Riemannian manifold. Assume that the Ricci curvature of $\mathcal{M}$ is *at least* as large as that of $\mathbb{S}^d$, a d-dimensional sphere. We then have $\operatorname{diam}(\mathcal{M}) \leqslant \operatorname{diam}\left(\mathbb{S}^d\right)$. Moreover, $\mathcal{M}$ is compact.

Part III: Notions of Curvature in Graphs

Graphs

In the following, we will be dealing with a graph $G = (V, E)$. We assume that the graph is connected. Graphs lack a 'smooth' structure, requiring a different treatment in terms of curvature.

Two curvature notions

1. Forman–Ricci curvature
2. Ollivier–Ricci curvature

Why is curvature useful?

Graph setting

Graph curvature explains problems in training GNNs. Negative curvature constitutes a bottleneck!

- J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong and M. M. Bronstein, 'Understanding over-squashing and bottlenecks on graphs via curvature', *International Conference on Learning Representations*, 2022, URL: `https://openreview.net/forum?id=7UmjRGzp-A`

Why is curvature useful?

Graph setting

Graph curvature explains problems in training GNNs. Negative curvature constitutes a bottleneck!

- J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong and M. M. Bronstein, 'Understanding over-squashing and bottlenecks on graphs via curvature', *International Conference on Learning Representations*, 2022, URL: `https://openreview.net/forum?id=7UmjRGzp-A`
- K. Nguyen, T. Nguyen, H. Nong, V. Nguyen, N. Ho and S. Osher, 'Revisiting Over-smoothing and Over-squashing using Ollivier–Ricci Curvature', Preprint, 2023, arXiv: `2211.15779 [cs.LG]`

Why is curvature useful?

Graph setting

Graph curvature explains problems in training GNNs. Negative curvature constitutes a bottleneck!

- J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong and M. M. Bronstein, 'Understanding over-squashing and bottlenecks on graphs via curvature', *International Conference on Learning Representations*, 2022, URL: `https://openreview.net/forum?id=7UmjRGzp-A`
- K. Nguyen, T. Nguyen, H. Nong, V. Nguyen, N. Ho and S. Osher, 'Revisiting Over-smoothing and Over-squashing using Ollivier–Ricci Curvature', Preprint, 2023, arXiv: `2211.15779 [cs.LG]`

Why is curvature useful?

Graph setting

Graph curvature explains problems in training GNNs. Negative curvature constitutes a bottleneck!

- J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong and M. M. Bronstein, 'Understanding over-squashing and bottlenecks on graphs via curvature', *International Conference on Learning Representations*, 2022, URL: `https://openreview.net/forum?id=7UmjRGzp-A`
- K. Nguyen, T. Nguyen, H. Nong, V. Nguyen, N. Ho and S. Osher, 'Revisiting Over-smoothing and Over-squashing using Ollivier–Ricci Curvature', Preprint, 2023, arXiv: `2211.15779 [cs.LG]`

Graph curvature also serves as a *characteristic property*, simplifying graph learning tasks.

- C. Coupette, S. Dalleiger and **B. Rieck**, 'Ollivier–Ricci Curvature for Hypergraphs: A Unified Framework', *International Conference on Learning Representations (ICLR)*, 2023, arXiv: `2210.12048 [cs.LG]`, in press

Why is curvature useful?

Graph setting

Graph curvature explains problems in training GNNs. Negative curvature constitutes a bottleneck!

- J. Topping, F. Di Giovanni, B. P. Chamberlain, X. Dong and M. M. Bronstein, 'Understanding over-squashing and bottlenecks on graphs via curvature', *International Conference on Learning Representations*, 2022, URL: `https://openreview.net/forum?id=7UmjRGzp-A`
- K. Nguyen, T. Nguyen, H. Nong, V. Nguyen, N. Ho and S. Osher, 'Revisiting Over-smoothing and Over-squashing using Ollivier–Ricci Curvature', Preprint, 2023, arXiv: `2211.15779 [cs.LG]`

Graph curvature also serves as a *characteristic property*, simplifying graph learning tasks.

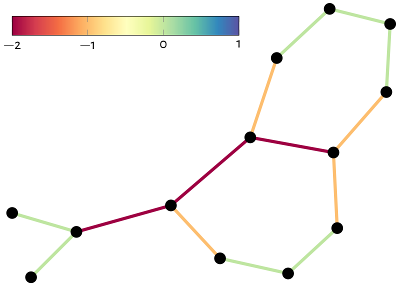
- C. Coupette, S. Dalleiger and **B. Rieck**, 'Ollivier–Ricci Curvature for Hypergraphs: A Unified Framework', *International Conference on Learning Representations (ICLR)*, 2023, arXiv: `2210.12048 [cs.LG]`, in press
- J. Southern*, J. Wayland*, M. Bronstein and **B. Rieck**, 'Curvature Filtrations for Graph Generative Model Evaluation', Preprint, 2023, arXiv: `2301.12906 [cs.LG]`

Forman–Ricci curvature

The *Forman–Ricci curvature* of an edge $(i, j) \in E$ is defined as

$$\kappa_{\mathrm{FR}}(i, j) := 4 - d_i - d_j + 3|\#_\Delta|, \tag{1}$$

where d_i is the degree of node i and $|\#_\Delta|$ is the number of 3-cycles (i.e. triangles) incident on i and j.



Ollivier–Ricci curvature

Let G be a graph with its shortest-path metric d and μ_v be a probability measure on G for node $v \in V$. The *Ollivier–Ricci curvature* of a pair of nodes $i \neq j \in V$ is then defined as

$$\kappa_{\mathrm{OR}}(i, j) := 1 - \frac{W_1(\mu_i, \mu_j)}{d(i, j)}, \tag{2}$$

where W_1 refers to the first *Wasserstein distance* between μ_i and μ_j.

Observation

This is, in some sense, the *natural generalisation* of Ricci curvature to a large class of objects.

History

First introduced by Ollivier[1] for metric (measure) spaces, this notion of curvature was quickly adopted for use in the graph setting.

[1] Y. Ollivier, 'Ricci curvature of Markov chains on metric spaces', *Journal of Functional Analysis* 256.3, 2009, pp. 810–864.

Useful properties of κ_{OR}

Lower bound

It is sufficient to know the values of κ_{OR} for each edge (i, j). If $\kappa_{OR}(i, j) \geqslant K$ for edges $(i, j) \in E$, then $\kappa_{OR}(k, l) \geqslant K$ for *all* pairs of vertices (k, l).

A Bonnet–Myers-like theorem

If $\kappa_{OR}(i, j) \geqslant K > 0$ for all edges $(i, j) \in E$, then for $i, j \in V$, we have

$$d(i, j) \leqslant \frac{W_1(\delta_i, \mu_i) + W_1(\delta_j, \mu_j)}{\kappa_{OR}(i, j)}, \tag{3}$$

where δ_i, δ_j refer to Dirac probability measures centred at node i and j, respectively. As a direct consequence, we obtain a *diameter bound* via

$$\operatorname{diam}(G) \leqslant \frac{\sup_i W_1(\delta_i, \mu_i)}{K}. \tag{4}$$

How to pick μ_i?

It is common practice to define a version of μ_i based on lazy random walks. Given a laziness parameter $\alpha \in [0, 1]$, we set

$$\mu_i(j) := \begin{cases} \alpha & \text{if } i = j \\ \frac{1-\alpha}{\deg(i)} & \text{if } i \neq j \text{ and } i \sim j \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where $\deg(i)$ refers to the degree of node i.

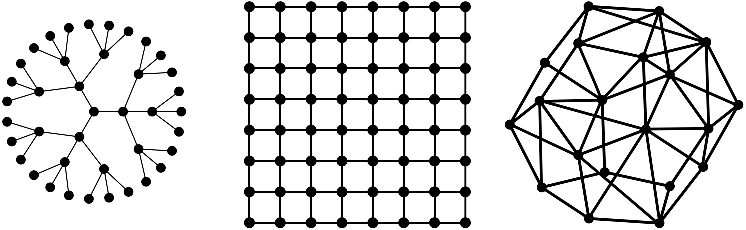
Observations

1. This is supposed to mimic the idea of curvature being a *local* property of a manifold.
2. With this definition, $\kappa_{OR}(i, j) \in [-2, 1]$ and $W_1(\delta_i, \mu_i) \leqslant 1$, leading to a Bonnet–Myers bound of $\operatorname{diam}(G) \leqslant \frac{2}{K}$.[2]

[2] Y. Lin, L. Lu and S.-T. Yau, 'Ricci curvature of graphs', *Tohoku Mathematical Journal* 63.4, 2011, pp. 605–627.

Ollivier–Ricci curvature

Canonical examples

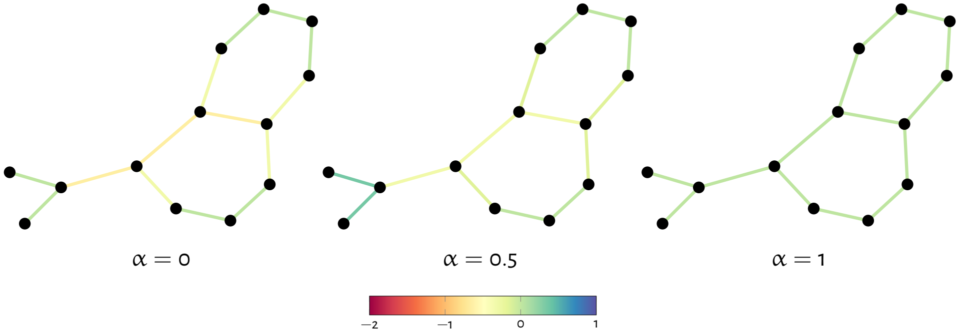


$\kappa_{OR} < 0$ $\kappa_{OR} = 0$ $\kappa_{OR} > 0$

(figure inspired by K. Devriendt and R. Lambiotte, 'Discrete curvature on graphs from the effective resistance', *Journal of Physics: Complexity* 3.2, 2022, p. 025008)

Ollivier–Ricci curvature

Examples



Part IV: Comparing Graph Generative Models

Problem

Given a *distribution* of graphs $\mathcal{G} = \{G_1, G_2, \ldots\}$, and different models for *generating* new graphs $\mathcal{G}' = \{G_1', G_2', \ldots\}$, how close (or similar) in the distributional sense are $\mathcal{G}$ and $\mathcal{G}'$?

Problem

Given a *distribution* of graphs $\mathcal{G} = \{G_1, G_2, \ldots\}$, and different models for *generating* new graphs $\mathcal{G}' = \{G_1', G_2', \ldots\}$, how close (or similar) in the distributional sense are $\mathcal{G}$ and $\mathcal{G}'$?

Applications

- Drug and molecule design
- Road networks
- Vessel networks

A new curvature-based comparison workflow

J. Southern*, J. Wayland*, M. Bronstein and **B. Rieck**, 'Curvature Filtrations for Graph Generative Model Evaluation', Preprint, 2023, arXiv: `2301.12906 [cs.LG]`

Joshua Southern

Jeremy Wayland

Michael Bronstein

Central premise

Since curvature is a multi-scale phenomenon, we need descriptors that are inherently capable of leveraging the multi-scale structure of a graph.

Proposed pipeline

We borrow ideas from *persistent homology* and use curvature as a filtration function of the graph. The advantage is that the resulting topological representations can be compared more easily.

Proposed pipeline

We borrow ideas from *persistent homology* and use curvature as a filtration function of the graph. The advantage is that the resulting topological representations can be compared more easily.

1. *Calculate* persistence diagrams based on a curvature filtration.

Proposed pipeline

We borrow ideas from *persistent homology* and use curvature as a filtration function of the graph. The advantage is that the resulting topological representations can be compared more easily.

1. *Calculate* persistence diagrams based on a curvature filtration.
2. *Convert* persistence diagrams into more suitable representations.

Proposed pipeline

We borrow ideas from *persistent homology* and use curvature as a filtration function of the graph. The advantage is that the resulting topological representations can be compared more easily.

1. *Calculate* persistence diagrams based on a curvature filtration.
2. *Convert* persistence diagrams into more suitable representations.
3. *Compare* representations (permutation tests, averages, distances, ...).

Computational topology

Filtrations

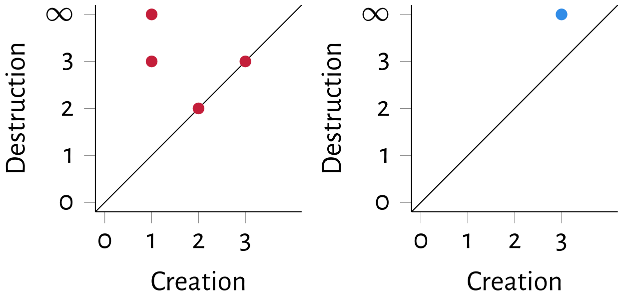
Given a scalar-valued function f on a graph, we obtain a natural *filtration* of the graph by analysing pre-images of f:

$f^{-1}((-\infty, 0])$ $f^{-1}((-\infty, 1])$ $f^{-1}((-\infty, 2])$ $f^{-1}((-\infty, 3])$

Computational topology

Persistence diagrams

We can calculate topological features—here, *connected components* and *cycles*—alongside the filtration, leading to multi-scale topological descriptors known as *persistence diagrams*:



Persistence diagrams form a metric space but also afford representations in Banach spaces as well as statistical analyses, making use of a transformation into *Betti curves* or *persistence landscapes*.

Results

Curvature filtrations are expressive

Success rate (↑) of distinguishing pairs of strongly-regular graphs when using either raw discrete curvature values, or a curvature filtration.

Data set	κ_{OR} (raw)	κ_{OR} (filtration)
`sr16622`	1.00	1.00
`sr261034`	0.78	0.89
`sr281264`	1.00	1.00
`sr361446`	0.00	0.02
`sr401224`	0.00	0.93

Results

Curvature filtrations help in substructure counting

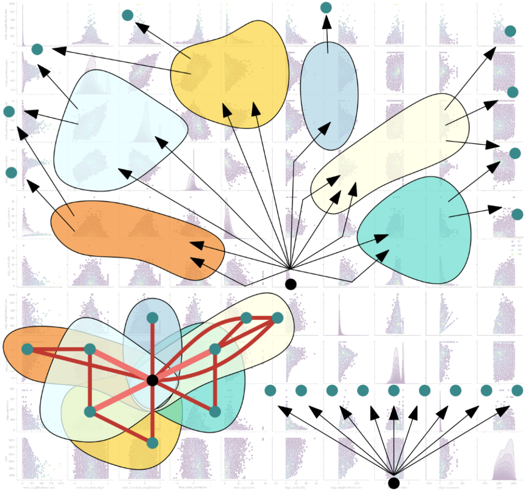
MAE (↓) for counting substructures based on raw curvature values and curvature-based filtrations. The Trivial Predictor always outputs the mean training target.

Method	Triangle	Tailed Triangles	Star	4-Cycle
Trivial Predictor	0.88	0.90	0.81	0.93
GCN	0.42	0.32	**0.18**	**0.28**
κ_{OR} (raw)	0.33	0.31	0.40	0.31
κ_{OR} (filtration)	**0.23**	**0.24**	0.34	0.31

Part V: Ollivier–Ricci Curvature for Hypergraphs

A framework for Ollivier–Ricci curvature on hypergraphs

C. Coupette, S. Dalleiger and **B. Rieck**, 'Ollivier–Ricci Curvature for Hypergraphs: A Unified Framework', *International Conference on Learning Representations (ICLR)*, 2023, arXiv: `2210.12048 [cs.LG]`, in press

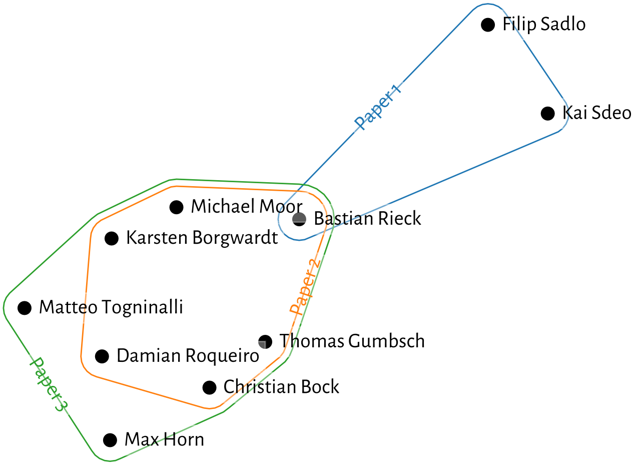


Corinna Coupette

Sebastian Dalleiger

Hypergraphs

A simple hypergraph $\mathrm{H} = (V, \mathrm{E})$ is a tuple of vertices V and hyperedges $\mathrm{E} \subseteq \mathcal{P}(V)$ (graphs are special cases of hypergraphs).



Hypergraphs capture higher-order relationships better than ordinary graphs.

How *should* curvature be generalised to hypergraphs?

For a pair of vertices (i, j) of a graph:

$$\kappa_{OR}(i, j) := 1 - \frac{W_1(\mu_i, \mu_j)}{d(i, j)}$$

Here, $d(i, j)$ refers to the shortest-path distance.

How *should* curvature be generalised to hypergraphs?

For a pair of vertices (i, j) of a graph:

$$\kappa_{OR}(i, j) := 1 - \frac{W_1(\mu_i, \mu_j)}{d(i, j)}$$

Here, $d(i, j)$ refers to the shortest-path distance.

For a hyperedge e, i.e. a set of vertices, of a hypergraph:

$$\kappa_{OR}(e) := 1 - \frac{\text{AGG}(e)}{d(e)}$$

Here, $d(e) := \max\{d(i, j) \mid \{i, j\} \subseteq e\}$, with $d(i, j)$ being the shortest-path distance.

How to choose aggregation functions?

Average aggregation

$$\textsc{Agg}_A(e) := \frac{2}{|e|(|e|-1)} \sum_{\{i,j\}\subseteq e} W_1(\mu_i, \mu_j) \tag{6}$$

How to choose aggregation functions?

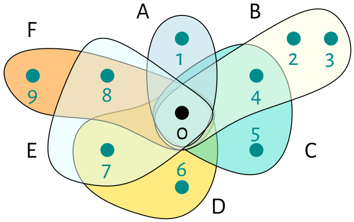
Average aggregation

$$\textsc{Agg}_{A}(e) := \frac{2}{|e|(|e|-1)} \sum_{\{i,j\} \subseteq e} W_1(\mu_i, \mu_j) \tag{6}$$

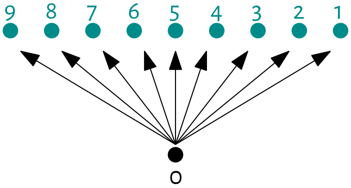
Maximum aggregation

$$\textsc{Agg}_{M}(e) := \max\left\{W_1(\mu_i, \mu_j) \mid \{i, j\} \subseteq e\right\} \tag{7}$$

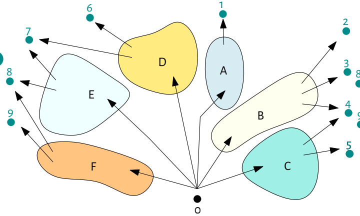
How to choose probability measures?



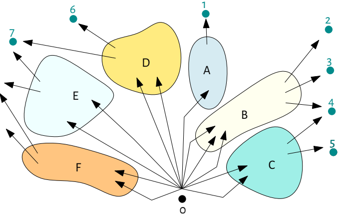
Neighbourhood



Equal-nodes



Equal-edges



Weighted-edges

Properties of hypergraph curvature

- Hypergraph curvature is negative/zero/positive for hypertrees/hypergrids/hypercliques, respectively.

Properties of hypergraph curvature

- Hypergraph curvature is negative/zero/positive for hypertrees/hypergrids/hypercliques, respectively.
- We obtain simple lower and upper bounds for the curvature of an edge in terms of the total variation distance between the probability measures.

Properties of hypergraph curvature

- Hypergraph curvature is negative/zero/positive for hypertrees/hypergrids/hypercliques, respectively.
- We obtain simple lower and upper bounds for the curvature of an edge in terms of the total variation distance between the probability measures.
- Hypergraph curvature relates local and global properties of a hypergraph.

Properties of hypergraph curvature

- Hypergraph curvature is negative/zero/positive for hypertrees/hypergrids/hypercliques, respectively.
- We obtain simple lower and upper bounds for the curvature of an edge in terms of the total variation distance between the probability measures.
- Hypergraph curvature relates local and global properties of a hypergraph.

Properties of hypergraph curvature

- Hypergraph curvature is negative/zero/positive for hypertrees/hypergrids/hypercliques, respectively.
- We obtain simple lower and upper bounds for the curvature of an edge in terms of the total variation distance between the probability measures.
- Hypergraph curvature relates local and global properties of a hypergraph.

Theorem

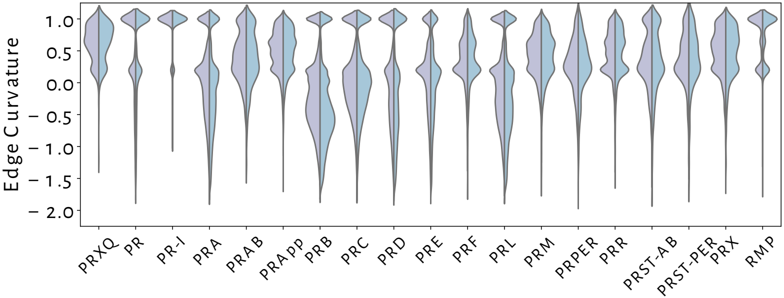
Given a subset of nodes $s \subseteq V$ *and an arbitrary probability measure* μ, *let* δ_i *denote a Dirac measure at node* i. *If*
(i) all curvatures based on μ *are strictly positive, i.e.,* $\kappa_{OR}(s) > 0$ *for all* $s \subseteq V$, *and*
(ii) $W_1(\mu_i, \mu_j) \leqslant \textsc{Agg}(s)$ *for* $\{i, j\} = \operatorname{argmax}(d(s))$, *then*

$$d(s) \leqslant \frac{W_1(\delta_i, \mu_i) + W_1(\delta_j, \mu_j)}{\kappa_{OR}(s)} \,. \tag{8}$$

Results

Hypergraph curvature is discriminative

Hypergraphs built on authorships of American Physical Society (APS) papers. Different collections (*journals*) exhibit distinct curvature patterns.

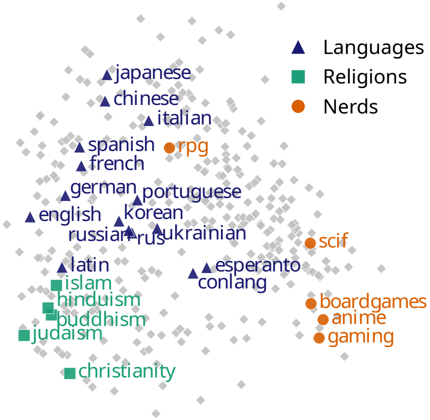


Using $\alpha = 0.1$. Left (violet): equal-edges; right (blue): weighted-edges

Results

Hypergraph curvature leads to interpretable embeddings

Hypergraphs built based on questions from `StackOverflow` forums (vertices: *tags*, edges: *questions*). Using an RBF kernel between distributions, we calculate embeddings via kernel PCA.



Discussion

Limitations

- Calculation does not scale well with increasing number of edges. Approximations required?

Discussion

Limitations

- Calculation does not scale well with increasing number of edges. Approximations required?
- Curvature is *oblivious* to attributes of the respective (hyper)graph.

Discussion

Limitations

- Calculation does not scale well with increasing number of edges. Approximations required?
- Curvature is *oblivious* to attributes of the respective (hyper)graph.
- So far, we are only working with very localised probability measures.

Discussion

Limitations

- Calculation does not scale well with increasing number of edges. Approximations required?
- Curvature is *oblivious* to attributes of the respective (hyper)graph.
- So far, we are only working with very localised probability measures.

Discussion

Limitations

- Calculation does not scale well with increasing number of edges. Approximations required?
- Curvature is *oblivious* to attributes of the respective (hyper)graph.
- So far, we are only working with very localised probability measures.

Summary

- Curvature is *expressive* and *useful* for graph learning tasks.

Discussion

Limitations

- Calculation does not scale well with increasing number of edges. Approximations required?
- Curvature is *oblivious* to attributes of the respective (hyper)graph.
- So far, we are only working with very localised probability measures.

Summary

- Curvature is *expressive* and *useful* for graph learning tasks.
- It provides a complementary multi-scale perspective on graphs and graph distributions.

Discussion

Limitations

- Calculation does not scale well with increasing number of edges. Approximations required?
- Curvature is *oblivious* to attributes of the respective (hyper)graph.
- So far, we are only working with very localised probability measures.

Summary

- Curvature is *expressive* and *useful* for graph learning tasks.
- It provides a complementary multi-scale perspective on graphs and graph distributions.
- Even in the more structured setting of graphs, there are several non-canonical choices to be made. We need to study their implications!