
graph_ricci_curvature

Release 1.0.0

Brian Andrews

Jul 01, 2024

CONTENTS:

1	graph_ricci_curvature	1
1.1	Installation	1
1.2	Usage	1
1.3	Manual	2
2	References	3
3	graph_ricci_curvature	5
3.1	graph_ricci_curvature package	5
4	Indices and tables	9
	Python Module Index	11
	Index	13

GRAPH_RICCI_CURVATURE

Calculate the Ricci curvature tensor for a networkx graph. Both Ollivier [1] and Forman [3] discretizations of Ricci curvature are implemented (see [4] for comparison of methods).

1.1 Installation

1.1.1 From Source

- Clone the repository and cd into the top level directory
- Install python's build: `python -m pip install build`
- Build the project: `python -m build`
- Install the project with pip: `python -m pip install dist/[file name].whl`

For testing the installation, you need pytest. Run pytest in the top level directory to run the full test suite.

1.1.2 Download the .whl from Releases

After download, install the wheel via pip: `python -m pip install [file name].whl`

1.1.3 From PyPi

Not done yet

1.2 Usage

After installation:

```
from graph_ricci_curvature.ollivier_ricci_curvature import OllivierRicciCurvature
import networkx as nx

#setting up a simple graph
G = nx.Graph()
G.add_nodes_from([1, 2, 3])
G.add_edges_from([(1, 2), (1, 3)])
```

(continues on next page)

(continued from previous page)

```
#create an object and calculate the values of the tensor and its contractions
g = OllivierRicciCurvature(G)
g.calculate_ricci_curvature()

#print results of calculation
print(list(g.G.edges.data()))
print(list(g.G.nodes.data()))
print(g.G.graph["graph_ricci_curvature"], g.G.graph["norm_graph_ricci_curvature"])
```

Output:

```
#edge curvature data
[
(1, 2, {"weight": 1.0, "ricci_curvature": 0.5}),
(1, 3, {"weight": 1.0, "ricci_curvature": 0.5}),
]

#node curvature data
[
(1, {"weight": 1.0, "ricci_curvature": 0.5}),
(2, {"weight": 1.0, "ricci_curvature": 0.5}),
(3, {"weight": 1.0, "ricci_curvature": 0.5})
]

#graph curvature data
1.5 0.5
```

1.3 Manual

You can see the manual [here](#) which is in docs/_build/latex. Or, after installation, can run the following with python

```
import graph_ricci_curvature as grc
grc.__manual__
```

to obtain the link to the pdf in the github repository.

REFERENCES

- [1] Ollivier, Y. 2009. “Ricci curvature of Markov chains on metric spaces”. *Journal of Functional Analysis*, 256(3), 810-864. DOI: <https://doi.org/10.1016/j.jfa.2008.11.001>, arXiv: <https://arxiv.org/abs/math/0701886>
- [2] Sandhu et al. 2015. “Graph Curvature for Differentiating Cancer Networks”. *Scientific Reports*. DOI: 10.1038/srep12323. DOI: <https://doi.org/10.1038/srep12323>.
- [3] R P Sreejith et al. “Forman curvature for complex networks” *J. Stat. Mech.* (2016) 063206. DOI: 10.1088/1742-5468/2016/06/063206. arXiv: <https://arxiv.org/pdf/1603.00386>.
- [4] Samal et al. “Comparative analysis of two discretizations of Ricci curvature for complex networks”. *Nature Scientific Reports*, 2018. <https://www.nature.com/articles/s41598-018-27001-3>.

GRAPH_RICCI_CURVATURE

3.1 graph_ricci_curvature package

3.1.1 Submodules

3.1.2 graph_ricci_curvature.forman_ricci_curvature module

References:

- [1] R P Sreejith et al J. Stat. Mech. (2016) 063206. DOI: 10.1088/1742-5468/2016/06/063206. arXiv: <https://arxiv.org/pdf/1603.00386>.

```
class graph_ricci_curvature.forman_ricci_curvature.FormanRicciCurvature(G: Graph,  
                                                                           edge_weight_key='weight',  
                                                                           node_weight_key='weight')
```

Bases: `_RicciCurvature`

Class for calculating Forman Ricci Curvature for a connected graph. Edge and node weights are set to 1.0 unless values are specified by the user in the input networkx graph object.

Parameters

G

[networkx graph] Input graph

edge_weight_key

[str] Key to specify edge weights in networkx graph. Default = weight.

node_weight_key

[str] Key to specify node weights in networkx graph. Default = weight.

calculate_edge_curvature(*source_node*, *target_node*)

Calculate value of Forman Ricci Curvature tensor associated with an edge between a source and target node defined as in References.

Parameters

source_node

[int or tuple] index of source_node in graph self.G

target_node

[int or tuple] index of target node in graph self.G

calculate_ricci_curvature(*norm=True*)

Calculate nonzero values of Ricci curvature tensor for all edges in graph self.G

Parameters

norm

[bool] If True, normalize nodal scalar curvature.

Returns

self.G

[networkx graph] Returns graph with ricci_curvature as graph, node, and edge attributes

3.1.3 graph_ricci_curvature.ollivier_ricci_curvature module

References:

- [1] Ollivier, Y. 2009. “Ricci curvature of Markov chains on metric spaces”. Journal of Functional Analysis, 256(3), 810-864. DOI: <https://doi.org/10.1016/j.jfa.2008.11.001>, arXiv: <https://arxiv.org/abs/math/0701886>
- [2] Sandhu et al. 2015. “Graph Curvature for Differentiating Cancer Networks”. Scientific Reports. DOI: 10.1038/srep12323. DOI: <https://doi.org/10.1038/srep12323>.

```
class graph_ricci_curvature.ollivier_ricci_curvature.OllivierRicciCurvature(G: Graph,  
                                                                           edge_weight_key='weight',  
                                                                           node_weight_key='weight')
```

Bases: `_RicciCurvature`

Class for calculating Ollivier Ricci Curvature of a connected graph. Only edge weights are considered in Ollivier curvature and are set to 1.0 if values are not provided in user or found in the input networkx graph object.

Parameters

G

[networkx graph] Input graph

edge_weight_key

[str] Key to specify edge weights in networkx graph. Default = weight.

node_weight_key

[str] Key to specify node weights in networkx graph. Default = weight.

calculate_edge_curvature(*source_node, target_node, alpha=0.5, dist_type='uniform', method='otd', weight_path_matrix=False, numThreads=1, reg=0.1*)

Calculate value of Ollivier Ricci Curvature tensor associated with an edge between a source and target node defined as

$1 - (\text{Wasserstein 1 Distance} / \text{Edge Weight})$

Parameters

source_node

[int or tuple] index of source_node in graph self.G

target_node

[int or tuple] index of target node in graph self.G

alpha

[float] hyperparameter ($0 \leq \alpha \leq 1$) determining how much mass to move from node

dist_type

[str] Distribution type for mass distribution in source or target node neighborhood. Default: uniform. Options: uniform, linear, inverse-linear, gaussian.

method

[str] Method for calculating optimal transport plan. Options: otd (optimal transport distance), sinkhorn

weight_path_matrix

[bool] When True, use edge weights when calculating shortest distance matrix. Default: False.

numThreads

[int] Specify number of threads for optimal transport plan. Only for “otd” method.

reg

[float] Regularization term to be used with “sinkhorn” method

Returns

curvature

[float] value of curvature tensor

calculate_ricci_curvature(*alpha=0.5, norm=True, dist_type='uniform', method='otd', weight_path_matrix=False, numThreads=1, reg=0.1*)

Calculate nonzero values of Ricci curvature tensor for all edges in graph self.G and tensor contractions.

Parameters

alpha

[float] Hyperparameter ($0 \leq \alpha \leq 1$) determining how much mass to move from node.

norm

[bool] If True, normalize nodal scalar curvature.

dist_type

[str] Distribution type for mass distribution in source or target node neighborhood. Default: uniform. Options: uniform, linear, inverse-linear, gaussian.

method

[str] Method for calculating optimal transport plan. Options: otd (optimal transport distance), sinkhorn.

weight_path_matrix

[bool] When True, use edge weights when calculating shortest distance matrix. Default: False.

numThreads

[int] Specify number of threads for optimal transport plan. Only for “otd” method.

reg

[float] Regularization term to be used with “sinkhorn” method.

Returns

self.G

[networkx graph] Returns graph with ricci_curvature as graph, node, and edge attributes

3.1.4 Module contents

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

g

`graph_ricci_curvature`, [8](#)
`graph_ricci_curvature.forman_ricci_curvature`,
[5](#)
`graph_ricci_curvature.ollivier_ricci_curvature`,
[6](#)

C

`calculate_edge_curvature()`
 (*graph_ricci_curvature.forman_ricci_curvature.FormanRicciCurvature*
 method), 5

`calculate_edge_curvature()`
 (*graph_ricci_curvature.ollivier_ricci_curvature.OllivierRicciCurvature*
 method), 6

`calculate_ricci_curvature()`
 (*graph_ricci_curvature.forman_ricci_curvature.FormanRicciCurvature*
 method), 6

`calculate_ricci_curvature()`
 (*graph_ricci_curvature.ollivier_ricci_curvature.OllivierRicciCurvature*
 method), 7

F

`FormanRicciCurvature` (class in
 graph_ricci_curvature.forman_ricci_curvature),
 5

G

`graph_ricci_curvature`
 module, 8

`graph_ricci_curvature.forman_ricci_curvature`
 module, 5

`graph_ricci_curvature.ollivier_ricci_curvature`
 module, 6

M

module
 graph_ricci_curvature, 8
 graph_ricci_curvature.forman_ricci_curvature,
 5
 graph_ricci_curvature.ollivier_ricci_curvature,
 6

O

`OllivierRicciCurvature` (class in
 graph_ricci_curvature.ollivier_ricci_curvature),
 6