graph_ricci_curvature

Release 0.1.0

Brian Andrews

Jun 27, 2024

CONTENTS:

CHAPTER

ONE

GRAPH_RICCI_CURVATURE

Calculate Ricci Curvature for a networkx graph

CHAPTER

TWO

INSTALLATION

2.1 From Source

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

2.2 Download the .whl from Releases

Not done yet

2.3 From PyPi

Not done yet

CHAPTER

THREE

USAGE

After installation:

```
from graph_ricci_curvature.ollivier_ricci_curvature import OllivierRicciCurvature
import networkx as nx
G = nx.Graph()
G.add_nodes_from([1, 2, 3])
G.add_edges_from([(1, 2), (1, 3)])
g = OllivierRicciCurvature(G)
g.calculate_ricci_curvature()
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:

```
[
(1, 2, {"weight": 1.0, "ricci_curvature": 0.5}),
(1, 3, {"weight": 1.0, "ricci_curvature": 0.5}),
]
[
(1, {'ricci_curvature': 0.5}),
(2, {'ricci_curvature': 0.5}),
(3, {'ricci_curvature': 0.5})
]
1.5 0.5
```

CHAPTER

FOUR

REFERENCES

- [1] https://www.nature.com/articles/s41598-018-27001-3

CHAPTER

FIVE

GRAPH_RICCI_CURVATURE

5.1 graph_ricci_curvature package

5.1.1 Submodules

5.1.2 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

5.1.3 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 - (Wasserstein 1 Distance / 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 <= alpha <=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 <= alpha <=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

5.1.4 Module contents

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

g

INDEX