HypercContagion Documentation

Release 0.1.2

Nicholas W. Landry and Joel C. Miller

January 12, 2023

HOME

For some introductory tutorials, see the Tutorials.

CHAPTER

ONE

SIM PACKAGE

Modules

epidemics	Classic epidemiological models extended to higher-order contagion.
opinions	Opinion formation models on hypergraphs.
functions	Provides predefined contagion functions for use in the hypercontagion library.

1.1 hypercontagion.sim.epidemics

Classic epidemiological models extended to higher-order contagion.

Functions

hypercontagion.sim.epidemics.**discrete_SIR**(*H*, *tau*, *gamma*, *transmission_function=<function threshold>*, *initial_infecteds=None*, *initial_recovereds=None*, *recovery_weight=None*, *transmission_weight=None*, *rho=None*, *tmin=0*, *tmax=inf*, *dt=1.0*, *return_event_data=False*, *seed=None*, ***args*)

Simulates the discrete SIR model for hypergraphs.

Parameters

- **H** (*xgi.Hypergraph*) – The hypergraph on which to simulate the SIR contagion process
- **tau** (*dict*) – Keys are edge sizes and values are transmission rates
- **gamma** (*float*) – Healing rate
- **transmission_function** (*lambda function, default: threshold*) – The contagion function that determines whether transmission is possible.
- **initial_infecteds** (*iterable, default: None*) – Initially infected node IDs.
- **initial_recovereds** (*iterable, default: None*) – Initially recovered node IDs.
- **recovery_weight** (*hashable, default: None*) – Hypergraph node attribute that weights the healing rate.
- **transmission_weight** (*hashable, default: None*) – Hypergraph edge attribute that weights the transmission rate.

- **rho** (*float, default: None*) – Fraction initially infected. Cannot be specified if *initial_infecteds* is defined.
- **tmin** (*float, default: 0*) – Time at which the simulation starts.
- **tmax** (*float, default: float("Inf")*) – Time at which the simulation terminates if there are still infected nodes.
- **dt** (*float, default: 1.0*) – The time step of the simulation.
- **return_event_data** (*bool, default: False*) – Whether to track each individual transition event that occurs.
- **seed** (*integer, random_state, or None (default)*) – Indicator of random number generation state.

Returns t, S, I, R

Return type tuple of np.arrays

Raises **HyperContagionError** – If the user specifies both rho and initial_infecteds.

hypercontagion.sim.epidemics.**discrete_SIS**(*H*, *tau*, *gamma*, *transmission_function=<function threshold>*, *initial_infecteds=None*, *recovery_weight=None*, *transmission_weight=None*, *rho=None*, *tmin=0*, *tmax=inf*, *dt=1.0*, *return_event_data=False*, *seed=None*, ***args*)

Simulates the discrete SIS model for hypergraphs.

Parameters

- **H** (*xgi.Hypergraph*) – The hypergraph on which to simulate the SIR contagion process
- **tau** (*dict*) – Keys are edge sizes and values are transmission rates
- **gamma** (*float*) – Healing rate
- **transmission_function** (*lambda function, default: threshold*) – The contagion function that determines whether transmission is possible.
- **initial_infecteds** (*iterable, default: None*) – Initially infected node IDs.
- **initial_recovereds** (*iterable, default: None*) – Initially recovered node IDs.
- **recovery_weight** (*hashable, default: None*) – Hypergraph node attribute that weights the healing rate.
- **transmission_weight** (*hashable, default: None*) – Hypergraph edge attribute that weights the transmission rate.
- **rho** (*float, default: None*) – Fraction initially infected. Cannot be specified if *initial_infecteds* is defined.
- **tmin** (*float, default: 0*) – Time at which the simulation starts.
- **tmax** (*float, default: float("Inf")*) – Time at which the simulation terminates if there are still infected nodes.
- **dt** (*float, default: 1.0*) – The time step of the simulation.
- **return_event_data** (*bool, default: False*) – Whether to track each individual transition event that occurs.
- **seed** (*integer, random_state, or None (default)*) – Indicator of random number generation state.

Returns t, S, I

Return type tuple of np.arrays

Raises **HyperContagionError** – If the user specifies both rho and initial_infecteds.

hypercontagion.sim.epidemics.**Gillespie_SIR**(*H*, *tau*, *gamma*, *transmission_function=<function threshold>*, *initial_infecteds=None*, *initial_recovereds=None*, *rho=None*, *tmin=0*, *tmax=inf*, *recovery_weight=None*, *transmission_weight=None*, *return_event_data=False*, *seed=None*, ***args*)

Simulates the SIR model for hypergraphs with the Gillespie algorithm.

Parameters

- **H** (*xgi.Hypergraph*) – The hypergraph on which to simulate the SIR contagion process
- **tau** (*dict*) – Keys are edge sizes and values are transmission rates
- **gamma** (*float*) – Healing rate
- **transmission_function** (*lambda function, default: threshold*) – The contagion function that determines whether transmission is possible.
- **initial_infecteds** (*iterable, default: None*) – Initially infected node IDs.
- **initial_recovereds** (*iterable, default: None*) – Initially recovered node IDs.
- **recovery_weight** (*hashable, default: None*) – Hypergraph node attribute that weights the healing rate.
- **transmission_weight** (*hashable, default: None*) – Hypergraph edge attribute that weights the transmission rate.
- **rho** (*float, default: None*) – Fraction initially infected. Cannot be specified if *initial_infecteds* is defined.
- **tmin** (*float, default: 0*) – Time at which the simulation starts.
- **tmax** (*float, default: float("Inf")*) – Time at which the simulation terminates if there are still infected nodes.
- **return_event_data** (*bool, default: False*) – Whether to track each individual transition event that occurs.
- **seed** (*integer, random_state, or None (default)*) – Indicator of random number generation state.

Returns t, S, I, R

Return type tuple of np.arrays

Raises **HyperContagionError** – If the user specifies both rho and initial_infecteds.

hypercontagion.sim.epidemics.**Gillespie_SIS**(*H*, *tau*, *gamma*, *transmission_function=<function threshold>*, *initial_infecteds=None*, *rho=None*, *tmin=0*, *tmax=100*, *recovery_weight=None*, *transmission_weight=None*, *return_event_data=False*, *seed=None*, ***args*)

Simulates the SIS model for hypergraphs with the Gillespie algorithm.

Parameters

- **H** (*xgi.Hypergraph*) – The hypergraph on which to simulate the SIR contagion process
- **tau** (*dict*) – Keys are edge sizes and values are transmission rates

- **gamma** (*float*) – Healing rate
- **transmission_function** (*lambda function, default: threshold*) – The contagion function that determines whether transmission is possible.
- **initial_infecteds** (*iterable, default: None*) – Initially infected node IDs.
- **initial_recovereds** (*iterable, default: None*) – Initially recovered node IDs.
- **recovery_weight** (*hashable, default: None*) – Hypergraph node attribute that weights the healing rate.
- **transmission_weight** (*hashable, default: None*) – Hypergraph edge attribute that weights the transmission rate.
- **rho** (*float, default: None*) – Fraction initially infected. Cannot be specified if *initial_infecteds* is defined.
- **tmin** (*float, default: 0*) – Time at which the simulation starts.
- **tmax** (*float, default: float("Inf")*) – Time at which the simulation terminates if there are still infected nodes.
- **return_event_data** (*bool, default: False*) – Whether to track each individual transition event that occurs.
- **seed** (*integer, random_state, or None (default)*) – Indicator of random number generation state.

Returns t, S, I

Return type tuple of np.arrays

Raises **HyperContagionError** – If the user specifies both rho and initial_infecteds.

hypercontagion.sim.epidemics.**event_driven_SIR**(*H*, *tau*, *gamma*, *transmission_function=<function majority_vote>*, *initial_infecteds=None*, *initial_recovereds=None*, *rho=None*, *tmin=0*, *tmax=inf*, *return_event_data=False*, *seed=None*, ***args*)

Simulates the SIR model for hypergraphs with the event-driven algorithm.

Parameters

- **H** (*xgi.Hypergraph*) – The hypergraph on which to simulate the SIR contagion process
- **tau** (*dict*) – Keys are edge sizes and values are transmission rates
- **gamma** (*float*) – Healing rate
- **transmission_function** (*lambda function, default: threshold*) – The contagion function that determines whether transmission is possible.
- **initial_infecteds** (*iterable, default: None*) – Initially infected node IDs.
- **initial_recovereds** (*iterable, default: None*) – Initially recovered node IDs.
- **recovery_weight** (*hashable, default: None*) – Hypergraph node attribute that weights the healing rate.
- **transmission_weight** (*hashable, default: None*) – Hypergraph edge attribute that weights the transmission rate.
- **rho** (*float, default: None*) – Fraction initially infected. Cannot be specified if *initial_infecteds* is defined.
- **tmin** (*float, default: 0*) – Time at which the simulation starts.

- **tmax** (*float, default: float("Inf")*) – Time at which the simulation terminates if there are still infected nodes.
- **return_event_data** (*bool, default: False*) – Whether to track each individual transition event that occurs.

Returns t, S, I, R

Return type tuple of np.arrays

Raises **HyperContagionError** – If the user specifies both rho and initial_infecteds.

hypercontagion.sim.epidemics.**event_driven_SIS**(*H*, *tau*, *gamma*, *transmission_function=<function majority_vote>*, *initial_infecteds=None*, *rho=None*, *tmin=0*, *tmax=inf*, *return_event_data=False*, *seed=None*, ***args*)

Simulates the SIS model for hypergraphs with the event-driven algorithm.

Parameters

- **H** (*xgi.Hypergraph*) – The hypergraph on which to simulate the SIR contagion process
- **tau** (*dict*) – Keys are edge sizes and values are transmission rates
- **gamma** (*float*) – Healing rate
- **transmission_function** (*lambda function, default: threshold*) – The contagion function that determines whether transmission is possible.
- **initial_infecteds** (*iterable, default: None*) – Initially infected node IDs.
- **initial_recovereds** (*iterable, default: None*) – Initially recovered node IDs.
- **recovery_weight** (*hashable, default: None*) – Hypergraph node attribute that weights the healing rate.
- **transmission_weight** (*hashable, default: None*) – Hypergraph edge attribute that weights the transmission rate.
- **rho** (*float, default: None*) – Fraction initially infected. Cannot be specified if *initial_infecteds* is defined.
- **tmin** (*float, default: 0*) – Time at which the simulation starts.
- **tmax** (*float, default: float("Inf")*) – Time at which the simulation terminates if there are still infected nodes.
- **return_event_data** (*bool, default: False*) – Whether to track each individual transition event that occurs.

Returns t, S, I

Return type tuple of np.arrays

Raises **HyperContagionError** – If the user specifies both rho and initial_infecteds.

1.2 hypercontagion.sim.opinions

Opinion formation models on hypergraphs.

Functions

hypercontagion.sim.opinions.**simulate_random_group_continuous_state_1D**(*H*, *initial_states*, *function=<function deffuant_weisbuch>*, *tmin=0*, *tmax=100*, *dt=1*, ***args*)

Simulate an opinion formation process where states are continuous and random groups are chosen.

Parameters

- **H** (*xgi.Hypergraph*) – the hypergraph of interest
- **initial_states** (*numpy array*) – initial node states
- **function** (*update function, default: deffuant_weisbuch*) – node update function
- **tmin** (*int, default: 0*) – the time at which the simulation starts
- **tmax** (*int, default: 100*) – the time at which the simulation terminates
- **dt** (*float > 0, default: 1*) – the time step to take.

Returns a 1D array of the times and a 2D array of the states.

Return type numpy array, numpy array

hypercontagion.sim.opinions.**simulate_random_node_and_group_discrete_state**(*H*, *initial_states*, *function=<function voter_model>*, *tmin=0*, *tmax=100*, *dt=1*, ***args*)

Simulate an opinion formation process where states are discrete and states are updated synchronously.

Parameters

- **H** (*xgi.Hypergraph*) – the hypergraph of interest
- **initial_states** (*numpy array*) – initial node states
- **function** (*update function, default: deffuant_weisbuch*) – node update function
- **tmin** (*int, default: 0*) – the time at which the simulation starts
- **tmax** (*int, default: 100*) – the time at which the simulation terminates
- **dt** (*float > 0, default: 1*) – the time step to take.

Returns a 1D array of the times and a 2D array of the states.

Return type numpy array, numpy array

hypercontagion.sim.opinions.**synchronous_update_continuous_state_1D**(*H*, *initial_states*, *function=<function hegselmann_krause>*, *tmin=0*, *tmax=100*, *dt=1*, ***args*)

Simulate an opinion formation process where states are continuous and states are updated synchronously.

Parameters

- **H** (*xgi.Hypergraph*) – the hypergraph of interest
- **initial_states** (*numpy array*) – initial node states
- **function** (*update function, default: deffuant_weisbuch*) – node update function
- **tmin** (*int, default: 0*) – the time at which the simulation starts
- **tmax** (*int, default: 100*) – the time at which the simulation terminates
- **dt** (*float > 0, default: 1*) – the time step to take.

Returns a 1D array of the times and a 2D array of the states.

Return type numpy array, numpy array

hypercontagion.sim.opinions.**hegselmann_krause**(*H*, *status*, *epsilon=0.1*)

The Hegselmann-Krause model.

Parameters

- **H** (*xgi.Hypergraph*) – the hypergraph of interest
- **status** (*iterable*) – statuses of the nodes.
- **epsilon** (*float, default: 0.1*) – confidence bound

Returns new opinions

Return type iterable

hypercontagion.sim.opinions.**deffuant_weisbuch**(*edge*, *status*, *epsilon=0.5*, *update='average'*, *m=0.1*)

the deffuant weisbuch model for updating the statuses of nodes in an edge

Parameters

- **edge** (*iterable*) – list of nodes
- **status** (*numpy array*) – node statuses
- **epsilon** (*float, default*) – confidence bound
- **update** (*str, default: "average"*) – if "average" the opinions of all nodes in the hyperedge are updated to the average. If "cautious", the nodes are moved toward the average.
- **m** (*float between 0 and 1, default: 0.1*) – the fraction of the possible distance to move the node opinions to the centroid.

Returns the updated statuses

Return type iterable

hypercontagion.sim.opinions.**discordance**(*edge*, *status*)

Computes the discordance of a hyperedge.

Parameters

- **edge** (*tuple*) – a list of an edge's members
- **status** (*numpy array*) – opinions of the nodes

Returns discordance of the hyperedge

Return type float

hypercontagion.sim.opinions.**voter_model**(*node*, *edge*, *status*, *p_adoption=1*)

the voter model given a hyperedge

Parameters

- **node** (*hashable*) – node whose opinion may change
- **edge** (*iterable*) – a list of the members of a hyperedge. must include the node.
- **status** (*dict*) – keys are node IDs, statuses are values
- **p_adoption** (*float, default: 1*) – probability that the node will adopt the consensus.

Returns new node status

Return type str

1.3 hypercontagion.sim.functions

Provides predefined contagion functions for use in the hypercontagion library.

Functions

hypercontagion.sim.functions.**collective_contagion**(*node*, *status*, *edge*)

Collective contagion function.

Parameters

- **node** (*hashable*) – node ID
- **status** (*dict*) – keys are node IDs and values are their statuses.
- **edge** (*iterable*) – hyperedge

Returns 0 if no transmission can occur, 1 if it can.

Return type int

hypercontagion.sim.functions.**individual_contagion**(*node*, *status*, *edge*)

Individual contagion function.

Parameters

- **node** (*hashable*) – node ID
- **status** (*dict*) – keys are node IDs and values are their statuses.
- **edge** (*iterable*) – hyperedge

Returns 0 if no transmission can occur, 1 if it can.

Return type int

hypercontagion.sim.functions.**threshold**(*node*, *status*, *edge*, *threshold=0.5*)

Threshold contagion process.

Contagion may spread if greater than a specified fraction of hyperedge neighbors are infected.

Parameters

- **node** (*hashable*) – node ID

- **status** (*dict*) – keys are node IDs and values are their statuses.
- **edge** (*iterable of hashables*) – nodes in the hyperedge
- **threshold** (*float, default: 0.5*) – the critical fraction of hyperedge neighbors above which contagion spreads.

Returns 0 if no transmission can occur, 1 if it can.

Return type int

hypercontagion.sim.functions.**majority_vote**(*node*, *status*, *edge*)

Majority vote contagion process.

Contagion may spread if the majority of a node's hyperedge neighbors are infected. If it's a tie, the result is random.

Parameters

- **node** (*hashable*) – node ID
- **status** (*dict*) – keys are node IDs and values are their statuses.
- **edge** (*iterable of hashables*) – nodes in the hyperedge

Returns 0 if no transmission can occur, 1 if it can.

Return type int

hypercontagion.sim.functions.**size_dependent**(*node*, *status*, *edge*)

CHAPTER

TWO

UTILITIES

Modules

utilities	Contains useful classes and functions for use in the hypercontagion library.

2.1 hypercontagion.utils.utilities

Contains useful classes and functions for use in the hypercontagion library.

Functions

hypercontagion.utils.utilities.**_process_trans_SIR_**(*t*, *times*, *S*, *I*, *R*, *Q*, *H*, *status*, *transmission_function*, *gamma*, *tau*, *source*, *target*, *rec_time*, *pred_inf_time*, *events*)

hypercontagion.utils.utilities.**_process_rec_SIR_**(*t*, *times*, *S*, *I*, *R*, *status*, *node*, *events*)

hypercontagion.utils.utilities.**_process_trans_SIS_**(*t*, *times*, *S*, *I*, *Q*, *H*, *status*, *transmission_function*, *gamma*, *tau*, *source*, *target*, *rec_time*, *pred_inf_time*, *events*)

hypercontagion.utils.utilities.**_process_rec_SIS_**(*t*, *times*, *S*, *I*, *status*, *node*, *events*)

CHAPTER

THREE

ABOUT

The HyperContagion library provides algorithms for simulating and visualizing contagion processes on complex systems with group (higher-order) interactions.

- Repository: https://github.com/nwlandry/hypercontagion
- PyPI: https://pypi.org/project/hypercontagion/
- Documentation: https://hypercontagion.readthedocs.io/

CHAPTER

FOUR

INSTALLATION

To install and use HyperContagion as an end user, execute

```
pip install hypercontagion
```

To install for development purposes, first clone the repository and then execute

```
pip install -e .['all']
```

If that command does not work, you may try the following instead

```
pip install -e .\[all\]
```

HyperContagion was developed and tested for Python 8-3.11 on Mac OS, Windows, and Ubuntu.

CHAPTER

FIVE

ACADEMIC REFERENCES

- The Why, How, and When of Representations for Complex Systems, Leo Torres, Ann S. Blevins, Danielle Bassett, and Tina Eliassi-Rad.
- Networks beyond pairwise interactions: Structure and dynamics, Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri.
- What are higher-order networks?, Christian Bick, Elizabeth Gross, Heather A. Harrington, Michael T. Schaub.

CHAPTER

SIX

CONTRIBUTING

If you want to contribute to this project, please make sure to read the code of conduct and the contributing guidelines.

The best way to contribute to HyperContagion is by submitting a bug or request a new feature by opening a new issue.

To get more actively involved, you are invited to browse the issues page and choose one that you can work on. The core developers will be happy to help you understand the codebase and any other doubts you may have while working on your contribution.

CHAPTER

SEVEN

CONTRIBUTORS

The core HyperContagion team members:

- Nicholas Landry
- Joel Miller

CHAPTER

EIGHT

LICENSE

PYTHON MODULE INDEX

h

INDEX

Symbols

C

D

E

G

H

I

M

S

T

V