SoSpaCATpy

Release 0.1.3

Jonas I. Liechti

Dec 11, 2020

CONTENTS:

1 Installation **3**

2 Example **5**

3 sospcat **7**

3.1 Submodules 8

3.1.1 sospcat.compare 8

3.1.2 sospcat.social 9

3.1.3 sospcat.spatial 9

4 Indices and tables **11**

Python Module Index **13**

Index **15**

This package is a python implementation of the Social-spatial Community Assignment Test (SoSpCAT).

SoSpCAT is a method to quantify the spatial signal present in social group structures.

CHAPTER

ONE

INSTALLATION

The package can be installed from the Python Package Index:

```
pip install sospcat
```

or directly from git with:

```
pip install git+https://github.com/j-i-l/sospcatpy.git
```

CHAPTER

TWO

EXAMPLE

to do

CHAPTER

THREE

SOSPCAT

The module exposes the functions `get_social_groups()` and `compare_soc_spat()`.

`get_social_groups()` is defined in `sospcat.social` as `spspcat.social.get_groups()`.

`compare_soc_spat()` is defined in `sospcat.compare` as `spspcat.social.social_vs_spatial()`.

compare_soc_spat(*node_locations*, *soc_membership*, ***kwargs*)

This method relies on the `homogeneity_completeness_v_measure()` method form the `sklearn.metrics` package.

See also:

V-Measure:
A conditional entropy-based external cluster evaluation measure
Andrew Rosenberg and Julia Hirschberg, 2007
http://acl.ldc.upenn.edu/D/D07/D07-1043.pdf

Parameters

- **node_locations** (*dict(int, tuple)*) – Specify for each node (key) its position (value) in the form of a (x, y) tuple.
- **soc_membership** (*dict*) – Holds for each node (key) the social group affiliation (value).
- ****kwargs** – All keyword arguments forwarded to `spatial.get_groups()`

Returns **homogeneity, completeness, v-measure** – Homogeneity, completeness and v-measure as computed by the `homogeneity_completeness_v_measure()`.

Return type tuple

get_social_groups(*a_graph*, *method='component_infomap'*, *return_form='membership'*)

Return the grouping of the provided graph object using the specified method. The grouping is returned as a list of sets each holding all members of a group.

Parameters

- **a_graph** (`igraph.Graph`) – The graph to partition
- **method** (*str (default='component_infomap')*) – String specifying which method to use. If two methods should be used one after the other they should be separated by _. Default: 'component_infomap' which will first consider all disconnected components as groups then apply infomap on all of those groups to optionally further split.

- **return_form** (*str (default='membership')*) – Determines the format of how the social group structure should be returned. Options are:
  - `'membership'`: A list returning for each *index* node the group it belongs to.
  - `'memberlists'`: Dictionary with a list of members *value* for each group *key*.

Returns

Depending on what was chosen for the *return_form* attribute, either the membership dict, i.e.:

```
{
    node_id: group_id,
    ...
}
```

or the memberlist dict, i.e.:

```
{
    group_id: [node1_id, node2_id, ...],
    ...
}
```

(value) is returned.

Return type dict

3.1 Submodules

3.1.1 sospcat.compare

Module holding functions for a quantitative comparison of groupings

social_vs_spatial (*node_locations*, *soc_membership*, ***kwargs*)

This method relies on the `homogeneity_completeness_v_measure()` method form the `sklearn.metrics` package.

See also:

V-Measure:
A conditional entropy-based external cluster evaluation measure
Andrew Rosenberg and Julia Hirschberg, 2007
http://acl.ldc.upenn.edu/D/D07/D07-1043.pdf

Parameters

- **node_locations** (*dict(int, tuple)*) – Specify for each node (key) its position (value) in the form of a (x, y) tuple.
- **soc_membership** (*dict*) – Holds for each node (key) the social group affiliation (value).
- ****kwargs** – All keyword arguments forwarded to *spatial.get_groups()*

Returns **homogeneity, completeness, v-measure** – Homogeneity, completeness and v-measure as computed by the `homogeneity_completeness_v_measure()`.

Return type tuple

3.1.2 sospcat.social

This module defines the methods for social community detection.

get_groups (*a_graph*, *method='component_infomap'*, *return_form='membership'*)

Return the grouping of the provided graph object using the specified method. The grouping is returned as a list of sets each holding all members of a group.

Parameters

- **a_graph** (`igraph.Graph`) – The graph to partition
- **method** (*str (default='component_infomap')*) – String specifying which method to use. If two methods should be used one after the other they should be separated by _. Default: 'component_infomap' which will first consider all disconnected components as groups then apply infomap on all of those groups to optionally further split.
- **return_form** (*str (default='membership')*) – Determines the format of how the social group structure should be returned. Options are:
  - `'membership'`: A list returning for each *index* node the group it belongs to.
  - `'memberlists'`: Dictionary with a list of members *value* for each group *key*.

Returns

Depending on what was chosen for the *return_form* attribute, either the membership dict, i.e.:

```
{
    node_id: group_id,
    ...
}
```

or the memberlist dict, i.e.:

```
{
    group_id: [node1_id, node2_id, ...],
    ...
}
```

(value) is returned.

Return type dict

3.1.3 sospcat.spatial

Module to define the spatial based partitioning

get_groups (*node_locations*, *nbr_clusters*, *return_form='membership'*, *random_state=None*, ***kwargs*)

Perform k-means clustering on the provided node locations.

Parameters

- **node_locations** (*dict*) – Specify for each node (key) its position (value) in the form of a tuple, *(x, y)*.
- **nbr_clusters** (*int*) – The number of clusters to find.
- **return_form** (*str (default='membership')*) – Determines the format of how the social group structure should be returned. Options are:
  - `'membership'`: A *dict* returning for each node (key) the group it belongs to (value).

  - `'memberlists'`: Dictionary with a list of members (value) for each group (key).

- **`random_state`** (*`int, RandomState instance, default=None`*) – Determines random number generation for centroid initialization. Use an int to make the randomness deterministic.

- **`**kwargs`** – All keyword arguments forwarded to `sklearn.cluster.KMeans`.

Returns

Depending on what was chosen for the *return_form* attribute, either the membership dict, i.e.:

```
{
    node_id: group_id,
    ...
}
```

or the memberlist dict, i.e.:

```
{
    group_id: [node1_id, node2_id, ...],
    ...
}
```

(value) is returned.

Return type dict

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

sospcat, 7
sospcat.compare, 8
sospcat.social, 9
sospcat.spatial, 9

INDEX

C

`compare_soc_spat()` (*in module sospcat*), 7

G

`get_groups()` (*in module sospcat.social*), 9
`get_groups()` (*in module sospcat.spatial*), 9
`get_social_groups()` (*in module sospcat*), 7

M

`module`
    `sospcat`, 7
    `sospcat.compare`, 8
    `sospcat.social`, 9
    `sospcat.spatial`, 9

S

`social_vs_spatial()` (*in module sospcat.compare*), 8
`sospcat`
    `module`, 7
`sospcat.compare`
    `module`, 8
`sospcat.social`
    `module`, 9
`sospcat.spatial`
    `module`, 9