MROCP Invariants API

API Documentation

November 10, 2014

Contents

1 Module computation.algs.degree.degree

1.1 Functions

r_igraph_degree(*g*, *mode*=`'total'`, *save_fn*=`None`)

Compute degree of graph g and save as necessary

Parameters

`g:` The igraph loaded via Rpy2 so an R object

`mode:` the type of degree. Default is an undirected i.e. in-degree + out degree

`save_fn:` the filename you want to use to save it. If not provided the graph adds a degree attribute to all nodes and returns.

Return Value

the graph with the degree attribute set.

1.2 Variables

Name	Description
__package__	**Value:** `'computation.algs.degree'`

2 Module computation.algs.eigen.eigen

2.1 Functions

r_igraph_eigs(*g*, *k*, *return_eigs*=`False`, *save_fn*=`None`, *real*=`True`, *lcc*=`False`)

Eigen spectral decomposition. Compute the top-k eigen pairs.

(section) Positional arguments

Parameters

- `g:` The igraph graph loaded via Rpy2 i.e. an R object
- `k:` the number of eigenpairs to compute. Must be < # nodes - 2
- `return_eigs:` boolean on whether to just return the eigenpairs or the whole graph
- `save_fn:` must an 2 item list/tuple with 2 names OR None
- `real:` Compute only the real part
- `lcc:` use the largest connected component only

Return Value

A graph with eigs as graph attributes OR actual eigenpairs

get_str_eigvects(*idx*)

Used for mapping to get eigenvectors that correspond to each vertex of the graph

Parameters

- `idx:` a 2-tuple that gives the indexes of the eigenvector 1-d flattened matrix that correspond to the particular vertex

Return Value

A vector i.e the eigenvector (latent position) for that vertex cast to a string

cut(*num*)

Shorten the format of a number to 2 decimal places plus exponent

Parameters

- `num:` the number to be shorten

2.2 Variables

Name	Description
gl_eigvects	**Value:** `None`
__package__	**Value:** `'computation.algs.eigen'`

3 Module computation.algs.scan1stat.scan1

3.1 Functions

r_igraph_scan1(*g*, *save_fn*=`None`)

Compute the scan statistic 1 of graph g and save as necessary

Parameters

- `g:` The igraph loaded via Rpy2 so an R object
- `save_fn:` the filename you want to use to save it. If not provided the graph adds a scan1 attribute to all nodes and returns.

Return Value

The graph with the scan1 attribute appended

3.2 Variables

Name	Description
__package__	**Value:** `'computation.algs.scan1stat'`

4 Module computation.algs.transitivity.transitivity

4.1 Functions

r_igraph_clust_coeff(*g*, *save_fn*=`None`)

Compute clustering coefficient/transitivity of graph g and save as necessary

Parameters

`g`: The igraph loaded via Rpy2 so an R object

`save_fn`: the filename you want to use to save it. If not provided the graph adds a clustcoeff attribute to all nodes and returns.

Return Value

the graph with the clustcoeff attribute appended

4.2 Variables

Name	Description
__package__	**Value:** `'computation.algs.transitivity'`

5 Module computation.algs.triangles.triangles

5.1 Functions

r_igraph_triangles(*g*, *save_fn*=`None`)

Compute local triangle count of graph g and save as necessary

Parameters

`g`: The igraph loaded via Rpy2 so an R object

`save_fn`: the filename you want to use to save it. If not provided the graph adds a tri count attribute to all nodes and returns.

Return Value

The graph with the tri vertex attribute appended

5.2 Variables

Name	Description
__package__	**Value:** `'computation.algs.triangles'`