CGraph documentation

Bruno Kim Medeiros Cesar

May 25, 2013

Abstract

Contents

1 `sorting`

2 `list`

3 `set`

4 `graph`

5 `graph_metric`

5.1 Constants

These constants are hard-coded to protect some numeric processes of hanging. They can be redefined during compilation, passing a flag such as `-DGRAPH_METRIC_TOLERANCE=1E-3`.

5.1.1 `GRAPH_METRIC_TOLERANCE`

Error tolerance for numeric methods.

5.1.2 `GRAPH_METRIC_MAX_ITERATIONS`

Maximum number of iterations for numeric methods.

5.2 Component identification and extraction

5.2.1 `graph_undirected_components`

Label vertices' components treating edges as undirected.

Preconditions `label` must have dimension n.

Postconditions `label[i]` is the component ID of vertex v_i.

Return Number of components

For directed graphs, considers adjacencies as incidences. Labels start from 0 and are sequential with step 1. Component IDs are not ordered according to size.

5.2.2 `graph_directed_components`

Label vertices' components treating edges as directed. NOT IMPLEMENTED YET.

Preconditions `label` must have dimension n.

Postconditions `label[i]` is the component ID of vertex v_i.

Return Number of components

For undirected graphs, simply call `graph_undirected_components`. For directed graphs, two vertices v_i and v_j are in the same component if and only if

$$d(v_i, v_j) \neq \infty$$
$$d(v_j, v_i) \neq \infty$$

where $d(u, v)$ is the geodesic distance between them. In other words, they are in the same component if they are mutually reachable.

Labels start from 0 and are sequential with step 1. Component IDs are not ordered according to size.

5.2.3 `graph_num_components`

Extract number of components from label vector.

Preconditions
$n > 0$
`label` must have dimension n.
`label` must contain sequential IDs starting from 0.

Return Number of components

5.2.4 `graph_components`

Map components to vertices from label vector.

Preconditions
$n > 0$
`label` must have dimension n.
`label` must contain sequential IDs starting from 0.
`comp` must have size `num_comp` and all sets should be already initialized.
`graph_num_components(g) == num_comp`

Postconditions
If v_i is in component c_j, then
`label[i] == j` and
`set_contains(comp[j], i)` is true.

Return Number of components

5.2.5 `graph_components`

Creates a new graph from `g`'s largest component.

The guarantee of vertices' order ID is the same as `graph_subset`. If two or more components have the same maximum size, one will be chosen in an undefined way.

Return A new graph isomorphic to `g`'s largest component.

Memory deallocation

```
graph_t *largest = graph_components(g);
delete_graph(largest);
```

5.3 Degree metrics

5.3.1 `graph_degree`

List all vertices' degrees.

Preconditions `degree` must have dimension n.

Postconditions `degree[i]` is the degree of vertex v_i.

The degree of a directed graph's vertex is defined as the sum of incoming and outgoing edges.

5.3.2 `graph_directed_degree`

List all vertices' incoming and outgoing degrees.

Preconditions

`g` must be directed. `in_degree` must have dimension n. `out_degree` must have dimension n.

Postconditions

`in_degree[i]` is the number of incoming edges to vertex v_i. `out_degree[i]` is the number of outgoing edges from vertex v_i.

5.4 Clustering metrics

5.4.1 `graph_clustering`

List all vertices' local clustering.

Preconditions

`g` must be undirected.
`clustering` must have dimension n.

Postconditions `clustering[i]` is the local clustering coefficient of vertex v_i.

The local clustering coefficient is only defined for undirected graphs, and gives the ratio of edges between a vertex' neighbors and all possible edges.

Formally,

$$C_i = \frac{e_i}{\binom{k_i}{2}} = \frac{2e_i}{k_i(k_i - 1)}$$

where

C_i is the local clustering coefficient of vertex v_i.

e_i is the number of edges between v_i's neighbors.

k_i is the degree of v_i.

If a vertex v_i has 0 or 1 adjacents, $C_i = 0$ by definition.

5.4.2 `graph_num_triplets`

Counts number of triplets and triangles (6 * number of closed triplets).

5.4.3 `graph_transitivity`

Compute the ratio between number of triangles and number of triplets.

5.5 Geodesic distance metrics

5.5.1 Definitions

5.5.2 `graph_geodesic_distance`

5.5.3 `graph_geodesic_vertex`

5.5.4 `graph_geodesic_all`

5.5.5 `graph_geodesic_distribution`

5.6 Centrality measures

5.6.1 `graph_betweenness`

5.6.2 `graph_eigenvector`

5.6.3 `graph_pagerank`

5.6.4 `graph_kcore`

5.7 Correlation measures

5.7.1 `graph_degree_matrix`

5.7.2 `graph_neighbor_degree_vertex`

5.7.3 `graph_neighbor_degree_all`

5.7.4 `graph_knn`

5.7.5 `graph_assortativity`