

Louvain Multilevel Refinement Community Detection

Description

The Louvain multilevel refinement community detection algorithm (Rotta & Noack, 2011) is an algorithm for performing community detection (clustering) in networks by maximizing a modularity function. The Louvain multilevel refinement algorithm can be used to detect communities in very large networks within short computing times. The algorithm extends the Louvain community detection algorithm proposed by Blondel et al. (2008) by offering a multilevel refinement procedure. This procedure tends to lead to higher modularity values, but it also increases the computing time that is required.

An alternatives to the Louvain multilevel refinement community detection algorithm is the smart local moving (SLM) community detection algorithm. This algorithms may yield even higher modularity values, but it may also require more computing time.

Adjust the following parameters to optimize the algorithm (See the paper for more detail):

Weight	An integer attribute of the edge that will be used as weight parameter
Modularity Function	The Standard modularity function has been proposed by Newman and Girvan (2004) and Newman (2004) . The Alternative modularity function has been proposed by Traag, Van Dooren, and Nesterov (2011)
Resolution	The resolution parameter determines the granularity level at which communities are detected. Use a value above (below) 1.0 if you want to obtain a larger (smaller) number of communities.
Random Start	Number of random starts
Iterations	Number of iterations per random start
Random Seed	Seed of the random number generator

Applications

The directionality of the input network does not matter, so both directed and undirected networks yield the same results.

If the input network has numeric edge attributes, one can be chosen as edge weight. If no edge weight (attribute) is specified, all edges default to having a weight of 1.

The output network will structurally be the same as the input network, but the nodes will be annotated with new attributes labeled "community_level_x", where x is a community level. The value of each of these attributes is the id of a community.

Implementation Details

A single network is expected as the input, and a single network is produced as the output.

A modified version of the Java implementation of this algorithm is compiled and wrapped for integration into CShell (see [Link](#) and [References](#)). This version is modified to accept input from method call rather than console.

To integrate this algorithm in CShell, a custom (Java) converter is used to convert the input network file to a edge list file that is proprietary to the compiled algorithm. The compiled algorithm is then executed upon this proprietary edge list file. The output community file is merged with the input network to produce the output network with annotations.

Usage Hints

The output of this algorithm can be visualized well with the Circular Hierarchy visualization or using Gephi.

Links

- Source Code: [Link](#)
- Original Java implementation: [Link](#)
- Paper: [Link](#)

References

1. Randolf Rotta, Andreas Noack, 2011. "Multilevel local search algorithms for modularity clustering". Journal of Experimental Algorithmics (JEA), Volume 16, 2011.

See Also



The license could not be verified: License Certificate has expired! [Generate a Free license now.](#)