Fast stochastic and recursive search algorithm

M. Rosvall*

Department of Physics, Umeå Univiversity, SE-901 87 Umeå, Sweden†

C. T. Bergstrom‡

Department of Biology, University of Washington, Seattle, WA 98195-1800 and Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501§

(Dated: March 13, 2009)

Any greedy (fast but inaccurate) or Monte Carlo-based (accurate but slow) approach can be used to minimize the map equation. But since on the order of 1000 networks must be clustered for each significance clustering and high accuracy always is desirable, we have developed a new method that provides a good balance between the two extremes. As a reference, the new algorithm is in practice as fast as the previous high-speed algorithms (the greedy search presented in the supporting appendix of ref. (1)), which were based on the method introduced in ref. (2) and refined in ref. (3). Yet it is also more accurate than our previous high-accuracy algorithm (a simulated annealing approach) presented in the same supporting appendix.

The core of the algorithm follows closely the method presented in ref. (4): neighboring nodes are joined into modules, which subsequently are joined into super modules and so on. First, each node is assigned to its own module. Then, in random sequential order, each node is moved to the neighboring module that results in the largest decrease of the map equation. If no move results in a decrease of the map equation, the node stays in its original module. This procedure is repeated, each time in a new random sequential order, until no move generates a decrease of the map equation. Now the network is rebuilt, with the modules of the last level forming the nodes at this level. And exactly as at the previous level, the nodes are joined into modules. This hierarchical rebuilding of the network is repeated until the map equation cannot be reduced further. Except for the random sequence order, this is the algorithm described in ref. (4).

With this algorithm, a fairly good clustering of the network can be found in a very short time. Let us call this the core algorithm and see how it can be improved. The often large number of nodes assigned to the same module are forced to move together once the network is rebuilt and what was an optimal move early in the algorithm might have the opposite effect later in the algorithm. Because two or more modules that merge together and form one single module when the network is rebuilt can never be separated again in this algorithm, the accuracy can be improved by breaking the modules of the final state of the core algorithm in either of the two following ways:

> *Submodule movements.* First each cluster is treated as a network on its own and the main algorithm is applied to this network. This procedure generates one or more submodules for each module. Then all submodules are moved back to their respective modules of the previous step. At this stage, with the same partition as in the previous step but with each submodule being freely movable between the modules, the main algorithm is re-applied.
>
> *Single-node movements.* First, each node is reassigned to be the sole member of its own module, in order to allow for single-node movements. Then all nodes are moved back to their respective modules of the previous step. At this stage, with the same partition as in the previous step but with each single node being freely movable between the modules, the main algorithm is re-applied.

In practice, we repeat the two extensions to the core algorithm in sequence and as long as the clustering is improved. Moreover, we apply the submodule movements recursively. That is, to find the submodules to be moved, the algorithm first splits the submodules into subsubmodules, subsubsubmodules, and so on until no further splits are possible. Finally, because the algorithm is stochastic and fast, we can restart the algorithm from scratch every time the clustering cannot be improved further and the algorithm stops. The implementation is straightforward and, by repeating the search 100 times, the final partition is less likely to correspond to a bad clustering of a local minimum. For each iteration, we record the clustering if the description length is shorter than the previously shortest description length.

*Electronic address: martin.rosvall@physics.umu.se
†URL: http://www.tp.umu.se/~rosvall
‡Electronic address: cbergst@u.washington.edu
§URL: http://octavia.zoology.washington.edu/

References

[1] M. Rosvall, C. T. Bergstrom, *PNAS* **105**, 1118 (2008).
[2] A. Clauset, M. E. J. Newman, C. Moore, *Phys Rev E* **70**, 066111 (2004).

[3] K. Wakita, T. Tsurumi, arXiv:cs/07020480 (2007).
[4] V. Blondel, J. Guillaume, R. Lambiotte, E. Mech, *J Stat Mech: Theory Exp* **2008**, P10008 (2008).