Orponen, Pekka; Schaeffer, Satu Elisa

Local clustering of large graphs by approximate Fiedler vectors

Published in:
4th International Workshop on Efficient and Experimental Algorithms, Santorini, Kreikka, 10.-13.5.2005

DOI:
10.1007/11427186_45

Published: 01/01/2005

Document Version
Early version, also known as pre-print

Please cite the original version:
Orponen, P., & Schaeffer, S. E. (2005). Local clustering of large graphs by approximate Fiedler vectors. In S. E. Nikoletseas (Ed.), *4th International Workshop on Efficient and Experimental Algorithms, Santorini, Kreikka, 10.-13.5.2005* (pp. 524-533). Springer. https://doi.org/10.1007/11427186_45

Local Clustering of Large Graphs by Approximate Fiedler Vectors [Extended Abstract]

Pekka Orponen* and Satu Elisa Schaeffer**

Laboratory for Theoretical Computer Science, P.O. Box 5400
FI-02015 TKK Helsinki University of Technology, Finland

Abstract We address the problem of determining the natural neighbourhood of a given node i in a large nonunifom network G in a way that uses only local computations, i.e. without recourse to the full adjacency matrix of G. We view the problem as that of computing potential values in a diffusive system where node i is fixed at zero potential, and the potentials at the other nodes are then induced by the adjacency relation of G. This point of view leads to a constrained spectral clustering approach. We observe that a gradient method for computing the respective Fiedler vector values at each node can be implemented in a local manner, leading to our eventual algorithm. The algorithm is evaluated experimentally using two types of nonuniform networks: randomised “caveman graphs” and a scientific collaboration network.

1 Introduction

The recent interest in the analysis of natural network data [14,20] has given rise to an array of fascinating algorithmic research issues. One key task is that of extracting natural *clusters* of nodes in a network that have a relatively high interconnectivity among themselves, and a relatively low connectivity to the rest of the network. (Also called “communities” in, e.g. [14,15].) Most of the existing literature on this topic considers the task of finding an ideal *global* clustering of a given graph. This is, however, infeasible by present techniques in the case of really large networks such as the WWW. For large networks, an effective clustering algorithm should scale at most linearly in the number of nodes n, whereas global clustering methods typically scale as $m \log m$ or mn, where m is the number of edges. In the case of the WWW, where n and m are currently in the order of several billions, such methods are quite inadequate. The fastest global algorithms can currently deal with networks containing up to maybe a few millions of nodes [11,14,15]. An added complication with online networks such as the WWW is that not all the nodes are directly accessible, and the graph can only be explored “on demand”.

In many applications it would in fact be sufficient to know the relevant cluster of a given source node, or maybe a group of nodes. Some recent papers, such as [18,21] address also this more limited goal. E.g. in [18], a parameter-free local clustering quality measure is optimised using simulated annealing; the computational effort needed to obtain the cluster of a given source node is quite modest and — most importantly — independent of the total size of the network,

* Research supported by the Academy of Finland, grant 204156. E-mail: pekka.orponen@tkk.fi

** Research supported by the Academy of Finland, grant 206235, and the Nokia Foundation. E-mail: elisa.schaeffer@tkk.fi

and the results seem to be quite robust with respect to variations in the annealing process.

One fascinating aspect of the clustering problem is that it is not even clear how the notion of a “natural cluster” of nodes in a graph should be defined. It is usually apparent to the human eye what the “correct” or at least “reasonable” clustering of a given node neighbourhood is, but this intuition is difficult to make precise in a way that could be reliably automated. The clustering quality measure in [18] is robust, easily computable and gives good results, but is somewhat heuristic. In the general literature, spectral and conductance-based notions are preferred on conceptual grounds [5,6,8,9,10,12,16,17], but are computationally demanding. (See, however, [13] for a distributed algorithm for decentralising the computational load.) Also flow-based and other more heuristic approaches have been proposed; see [1,7] for overviews and comparisons.

In [21] the clustering task is formulated, with the goal of efficient computation, as a problem of determining voltage levels in an electrical circuit with unit resistances corresponding to the edges of the original network. The source node is fixed at a high potential and a randomly selected target node at low potential; an approximate solution to the Kirchhoff equations is computed by an iteration scheme, and the eventual cluster of the source node is deemed to consist of those nodes whose voltages are “close” to the high value. The possibility that the target node is accidentally selected from within the natural cluster of the source node is decreased by repeating the experiment some small number of times and determining cluster membership by majority vote.

This electrical circuit analogue appears to have been first suggested in [15], where however the aim is to compute a global clustering of a given network by considering all possible source-target pairs, and for each pair solving the Kirchhoff equations exactly by explicitly inverting the corresponding Laplacian matrix. (We note that since solutions of the Kirchhoff equations can be decomposed in terms of the eigenvectors of the circuit graph Laplacian, this method is actually also a variant of the spectral partitioning techniques.)

In Section 2, we present our local clustering algorithm, which improves on [15,21] by eliminating the need for arbitrary “target” nodes, and by making the connection to spectral methods explicit. Section 3 discusses our experiments with the method. Section 4 summarises the work and addresses directions for further research.

2 Approximate computation of Fiedler vectors

We continue the analogue of representing cluster membership values as physical potentials, but eliminate the unnatural choice of random “target” nodes by basing our model on diffusion in an *unbounded* medium rather than the electrical closed-circuit model. Thus, given a graph G and a source node i, we fix i at a constant potential level, which we choose to be zero, and consider the solution to the discrete Dirichlet problem on G with this single-node boundary condition [2, p. 128]. For clustering purposes, we find an eigenvector u corresponding to the smallest eigenvalue σ_1 of the respective Dirichlet matrix, i.e. the Laplacian matrix of G with row and column i removed [2,3]. This eigenvector u, the *(Dirichlet-)Fiedler* vector of G, will now assign potential values $u(j)$ close to 0 for nodes j that are within a densely interconnected neighbourhood of the source node i, and larger values for nodes that have sparser connections to the source. The method obviously generalises to starting from a larger set of source nodes, if desired.

Since we wish to develop a local algorithm, and not deal with the full adjacency matrix of the network, we approach the computation of the Fiedler vector u via minimising the Rayleigh quotient [2,3]:

$$\sigma_1 \quad = \quad \inf_u \frac{\sum_{j\sim k}(u(j)-u(k))^2}{\sum_j u(j)^2}, \tag{1}$$

where the infimum is computed over vectors u satisfying the boundary condition $u(i) = 0$ at the source node(s). (The notation $j \sim k$ is an abbreviation for $(j,k) \in E$.) Furthermore, since we are free to normalise our eventual Fiedler vector to any length we wish, we can constrain the minimisation to vectors u that satisfy, say, $\|u\|_2^2 = n = |V|$. Thus, the task becomes one of finding a vector u that satisfies:

$$u \quad = \quad \text{argmin}\left\{ \sum_{j\sim k}(u(j)-u(k))^2 \;\middle|\; u(i)=0,\ \|u\|_2^2 = n \right\}. \tag{2}$$

We can solve this task approximately by reformulating the requirement that $\|u\|_2^2 = n$ as a "soft constraint" with weight $c > 0$, and minimising the objective function

$$f(u) \quad = \quad \frac{1}{2}\sum_{j\sim k}\Big(u(j)-u(k)\Big)^2 + \frac{c}{2}\cdot\Big(n - \sum_j u(j)^2\Big) \tag{3}$$

by gradient descent. Since the partial derivatives of f have the simple form

$$\frac{\partial f}{\partial u(j)} \quad = \quad -\sum_{k\sim j} u(k) + (\deg(j) - c)\cdot u(j), \tag{4}$$

the descent step can be computed locally at each node, based on information about the u-estimates at the node itself and its neighbours:

$$\tilde{u}_{t+1}(j) \quad = \quad \tilde{u}_t(j) + \delta\cdot(\sum_{k\sim j}\tilde{u}(k) - (\deg(j)-c)\cdot\tilde{u}(j)), \tag{5}$$

where $\delta > 0$ is a parameter determining the speed of the descent.

Assuming that the natural cluster of node i is small compared to the size of the full network, the normalisation $\|u\|_2^2 = n$ entails that most nodes j in the network will have $u(j) \approx 1$. Thus the descent iterations (5) can be started from an initial vector $\tilde{u}_0$ that has $\tilde{u}_0(i) = 0$ for the source node i and $\tilde{u}_0(k) = 1$ for all $k \neq i$. The estimates need then to be updated at time $t > 0$ only for those nodes j that have neighbours $k \sim j$ such that $\tilde{u}_{t-1}(k) < 1$.

Balancing the constraint weight c against the speed of gradient descent δ naturally requires some care. We have obtained reasonably stable results with the following heuristic: given an estimate $\bar{k}$ for the average degree of the nodes in the network, set $c = 1/\bar{k}$ and $\delta = c/10$. The gradient iterations (5) are then continued until all the changes in the u-estimates are below $\varepsilon = \delta/10$. The

$\tilde{u(j)}$ values are thresholded at 1, so that if the right hand side of equation (5) suggests a value greater than this, then a value of 1 is used in the update instead. Occasionally equation (5) may suggest also negative u-estimates, but this we have taken as an indication of a too rapid descent, and have restarted the run with a smaller value of δ.

The eventual (approximate) Fiedler values thus represent the degree of membership of each node j in the cluster of node i. A fully automated clustering system needs to still determine a good cluster boundary for node i, based on these values. This is a simple one-dimensional two-classification task that can in principle be solved using any of the standard pattern classifiers, such as the k-means algorithm [4]. However since we wish to maintain the locality of our method also at this stage, the most obvious implementations of these algorithms are not acceptable to us. (We have not yet looked into the possibility of localising the standard classifiers.) One simple local approach would be to just threshold the potentials as in [21], but we prefer not to introduce any additional instance-specific parameters to the algorithm.

Rather, we choose to follow the approach of [18,19] of defining a locally computable cluster quality measure and optimising it by some local process — currently by a simulated annealing computation that modifies (expands or contracts) a candidate cluster one node at a time, with a time-increasing preference towards modifications that improve cluster quality. Given a source node i and a candidate cluster S containing i, a natural family of quality measures is provided by the *weighted Cheeger ratios* [2, p. 35]:

$$h_w(S) \quad = \quad \frac{\sum_{j \in S} \sum_{k \sim j, k \notin S} w(j,k)}{\sum_{j \in S} \sum_{k \sim j} w(j,k)}, \tag{6}$$

where $w(j,k)$ is an appropriate nonnegative edge weight function. Clusters S with low Cheeger ratios have low (weighted) extracluster connectivity, and high (weighted) intracluster connectivity, as is to be intuitively expected of a good cluster. Thus, aiming to minimise this ratio seems like a reasonable thing to do, and is also justified by general isoperimetric principles. In our experiments, edge weights determined as $w(j,k) = (|u(j) - u(k)|)^{-1}$ seem to lead to natural clusters in different types of networks, and are also intuitively appealing.

3 Experiments

We report on tests of our local Fiedler clustering method on two types of networks: randomised "caveman graphs" with 138 and 1533 nodes, and a "collaboration graph" representing a network of 503 mathematicians and computer scientists and their pairwise coauthorships.

The synthetic caveman graphs (cf. Figure 1) were generated according to a probabilistic variation of the deterministic construction given in [20, p. 103]. Whereas the recipe in [20] stipulates that a caveman graph of size $n = rk$ and cavesize k consist of exactly r copies of a k-clique connected together into a cycle in a specific way, our construction gives only probabilistic parameters for the expected size, number and connection densities of the caves, resulting in a somewhat more natural family of test graphs with nevertheless predictable

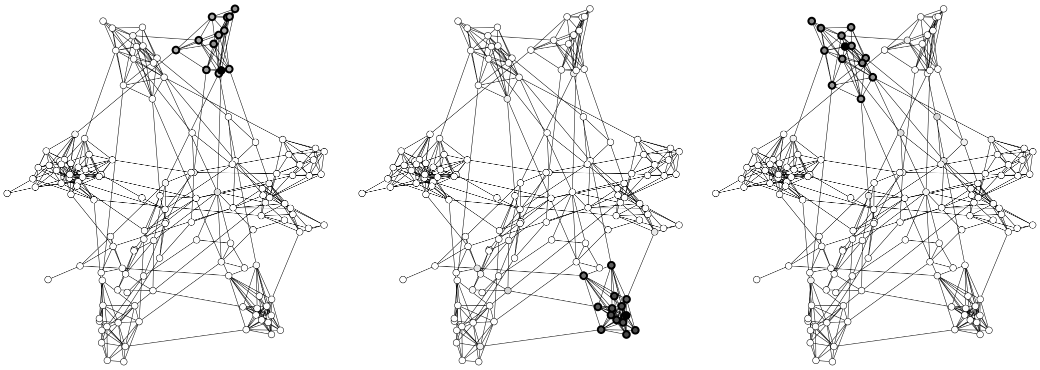

Figure 1. Local Fiedler clusters in a 138-node caveman graph.

clustering properties. (The precise graph generation method is given in [19, p. 94].)

Figure 1 represents the results of the approximate Fiedler vector calculations on the 138-node caveman graph, starting from three different source nodes. For visual effect, the nodes are colour-coded so that dark colours correspond to small approximated Fiedler potential values, with the source node in each case coloured black. The parameter values used in this case were the standard ones derived from $\bar{k} = 5.1$ (i.e. $c = 0.20$, $\delta = 0.002$, $\varepsilon = 0.0002$). As can be seen, the method discerns the natural clusters embedded in the graph quite distinctly. The nodes selected by the Cheeger ratio heuristic for the relevant clusters in each of the three cases are indicated by thickened node boundaries; also the clusters determined in this manner can be seen to correspond to the natural ones. (The smaller, 138-node graph was chosen here merely for illustrative purposes. The results on the bigger, 1533-node graph are qualitatively similar, but the graph is too large to be represented in a drawing.)

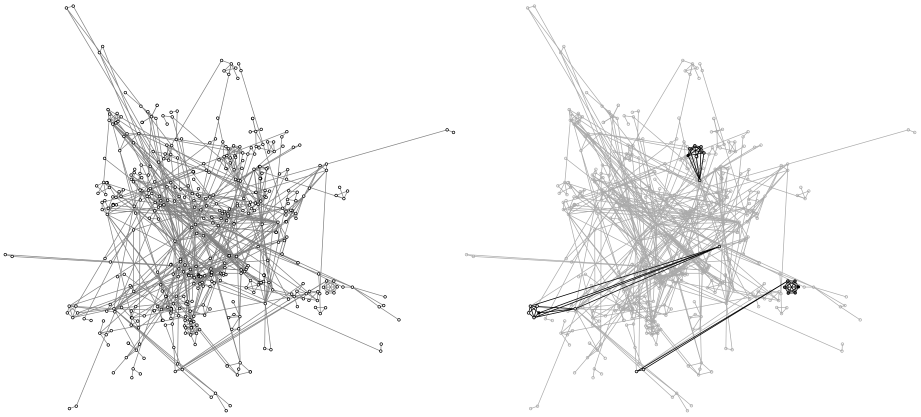

Figure 2. Local Fiedler clusters in a 503-node collaboration graph.

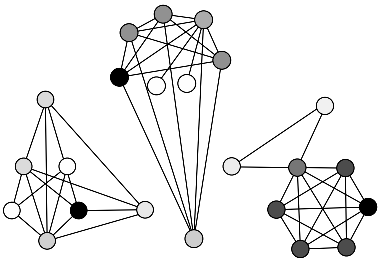

Figure 3. A closeup view of the three clusters.

Our other test graph was extracted from the Mathematics section of the Karlsruhe Collection of Computer Science Bibliographies.[1] The raw coauthorship data was cleaned in various ways to eliminate nonperson authors such as institutes and committees, unify spellings of authors' names, etc. (details given in [19, p. 99]). The resulting 503-node graph is shown in the left panel of Figure 3. The right panel shows three small collaborative clusters identified by the local Fiedler clustering method, starting from three distinct source nodes; the clusters are non-overlapping in the sense that none of the source nodes gave values less than 1.0 for any of the members of the other two clusters. Figure 3 presents a close-up view of the three clusters indicated in Figure 3, with distant and overlapping nodes rearranged to allow a better view of the structure of the induced subgraphs. Also in this instance, our standard parameter values based on $\bar{k} = 3.3$ (i.e. $c = 0.30$, $\delta = 0.03$, $\varepsilon = 0.003$) were used.

We wish to emphasize that the small size of our example graphs here is due to the requirements of illustration. The fact that our method is *local* means exactly that its running time scales relative to the size of the resulting *cluster*, and does *not* depend on the size of the ambient graph.

In fact, we have also implemented the method in such a manner that the u-estimates are updated according to equation (5) only as required by the optimisation process of the Cheeger clustering criterion (6). This means, firstly, that nodes that fall out of the single-edge neighbourhood of an evolving candidate cluster no longer need to be accessed, and secondly, that nodes that remain in the cluster have their u-estimates updated repeatedly in connection with reevaluations of the Cheeger criterion, thus implicitly focusing the gradient descent of the u-estimates to the part of the graph that is of interest for the clustering goal. This implementation saves considerably in both the working space and the running time requirements of the algorithm, without affecting the quality of the results.

4 Conclusions and Further Work

We presented a local method for clustering graphs based on computing their approximate Fiedler vectors and illustrated its behaviour on simple "caveman" and "collaboration" graphs. According to our experiments, the method behaves well and conforms to the intuition that arises from its analytical properties. The

[1] `http://liinwww.ira.uka.de/bibliography/Math/`, accessed 2 December 2002.

key characteristic of the method is that its resource requirements depend only on the size and connectivity of the resulting cluster, and *not* on the characteristics of the whole graph.

As future work, the algorithm should also be extended to work on directed graphs, in order to deal with interesting natural networks such as the WWW. Some interesting issues remain also in the area of localising standard clustering methods and comparing them to the presently used Cheeger criterion optimisation technique.

Acknowledgments

We thank most appreciatively Mr. Kosti Rytkönen for providing us with his automatic graph drawing tool used to produce the diagrams in Figures 1 and 3..

References

1. U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering algorithms. In *Proceedings of the 11th Annual European Symposium on Algorithms(ESA'03), Lecture Notes in Computer Science 2382*, pages 568–579, Berlin Heidelberg, 2003. Springer-Verlag.
2. F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, Providence, RI, 1997.
3. F. R. K. Chung and R. B. Ellis. A chip-firing game and Dirichlet eigenvalues. *Discrete Mathematics*, 257:341–355, 2002.
4. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, NY, 2001.
5. M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23:298–305, 1973.
6. M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25:619–633, 1975.
7. G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of Web communities. *IEEE Computer*, pages 66–71, 2002.
8. C. Gkantsidis, M. Mihail, and E. Zegura. Spectral analysis of Internet topologies. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, pages 364–374, New York, NY, 2003. IEEE.
9. S. Guattery and G. L. Miller. On the quality of spectral separators. *SIAM Journal on Matrix Analysis and Applications*, 19(3):701–719, 1998.
10. X. He, H. Zha, C. H. Q. Ding, and H. Simon. Web document clustering using hyperlink structures. *Computational Statistics & Data Analysis*, 41(1):19–45, 2002.
11. J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 541–546, New York, NY, 2003. ACM.
12. R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.
13. D. Kempe and F. McSherry. A decentralized algorithm for spectral analysis. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC'04)*, New York, NY, 2004. ACM.
14. M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066113, 2004.
15. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.
16. A. Pothen, H. D. Simon, and K. P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal of Matrix Analysis and Applications*, 11:430–452, 1990.
17. D. A. Spielman and S.-H. Teng. Spectral partitioning works: planar graphs and finite element meshes. In *Proceedings of the 37th IEEE Symposium on Foundations of Computing (FOCS'96)*, pages 96–105, Los Alamitos, CA, 1996. IEEE Computer Society.
18. S. E. Virtanen. Clustering the Chilean Web. In *Proceedings of the First Latin American Web Congress*, pages 229–231, Los Alamitos, CA, 2003. IEEE Computer Society.

19. S. E. Virtanen. Properties of nonuniform random graph models. Research Report A77, Helsinki University of Technology, Laboratory for Theoretical Computer Science, Espoo, Finland, May 2003. URL: `http://www.tcs.hut.fi/Publications/info/bibdb.HUT-TCS-A77.shtml`.
20. D. J. Watts. *Small Worlds: The Dynamics of Networks between Ordeer and Randomness*. Princeton University Press, Princeton, RI, 1999.
21. F. Wu and B. A. Huberman. Finding communities in linear time: a physics approach. *The European Physics Journal B*, 38:331–338, 2004.