Stochastic Local Clustering for Massive Graphs

Satu Elisa Schaeffer

Helsinki University of Technology, Espoo, Finland
elisa.schaeffer@tkk.fi
http://www.tcs.hut.fi/~satu/

Abstract. Most graph-theoretical clustering algorithms require the complete adjacency relation of the graph representing the examined data. This is infeasible for very large graphs currently emerging in many application areas. We propose a local approach that computes clusters in graphs, one at a time, relying only on the neighborhoods of the vertices included in the current cluster candidate. This enables implementing a local and parameter-free algorithm. Approximate clusters may be identified quickly by heuristic methods. We report experimental results on clustering graphs using simulated annealing.

1 Introduction

Many practical applications of information processing involve massive amounts of data, much of which tends to be noise, and only a small fraction contains semantically interesting information. In general, *clustering* is the process of organizing such data into meaningful groups in order to interpret properties of the data. Some general clustering methods operate online [2, 9], but practically all existing methods require some parameters in addition to a distance measure, such as the number of clusters to produce. In graph clustering, the data consists of a set of n *vertices* V connected by a set of m *edges* E. A *cluster* in a graph $G = (V, E)$ is considered to be a set of vertices that have relatively many connections among themselves with respect to the graph structure on a global scale. The existing methods are mostly global and rely on the full adjacency relation of the graph or a derived measure, such as the graph's eigenvalue spectrum [3].

2 Local Clustering

For many applications, only a small subset of vertices needs to be clustered instead of the whole graph. These include locating documents or genes closely related to a given "seed" data set. This motivates the use of a *local* approach for finding a good cluster containing a specified vertex or a set of vertices by examining only a limited number of vertices at a time, proceeding in the "vicinity" of the seed vertex. The scalability problem is avoided, as the graph as a whole never needs to be processed and clusters for different seeds may be obtained by parallel computation.

T.B. Ho, D. Cheung, and H. Liu (Eds.): PAKDD 2005, LNAI 3518, pp. 354–360, 2005.

We adopt the following notation: in a graph $G = (V, E)$, a cluster candidate is a set of vertices $\mathcal{C} \subseteq V$. The *order* of the cluster is the number of vertices included in the cluster, denoted by $|\mathcal{C}|$. Two vertices u and v are said to be *neighbors* if $(u, v) \in E$. Following common criteria [3, 5, 8], we want the clusters to be vertex sets that are connected in G by many internal connections and only few connections outside. We define the *internal degree* of a cluster $\mathcal{C}$ to be the number of edges connecting vertices in $\mathcal{C}$ to each other:

$$\deg_{\text{int}}(\mathcal{C}) = |\{(u, v) \in E \mid u, v \in \mathcal{C}\}|. \tag{1}$$

The *local density*[1] of a cluster $\mathcal{C}$ is

$$\delta_\ell(\mathcal{C}) = \frac{\deg_{\text{int}}(\mathcal{C})}{\binom{|\mathcal{C}|}{2}} = \frac{2 \deg_{\text{int}}}{|\mathcal{C}|(|\mathcal{C}| - 1)}. \tag{2}$$

Clearly, optimizing $\delta_\ell \in [0, 1]$ alone makes small cliques superior to larger but slightly sparser subgraphs, which is often impractical. For clusters to have only a few connections to the rest of the graph, one may optimize the *relative density* $\delta_r(\mathcal{C})$ (see [6] and the references therein); it is defined in terms of the internal degree $\deg_{\text{int}}$ (Equation 1) and *external degree* of a cluster $\mathcal{C}$,

$$\deg_{\text{ext}}(\mathcal{C}) = |\{(u, v) \in E \mid u \in \mathcal{C}, v \in V \setminus \mathcal{C}\}|, \tag{3}$$

as the ratio of the internal degree to the number of edges incident to the cluster,

$$\delta_r(\mathcal{C}) = \frac{\deg_{\text{int}}(\mathcal{C})}{\deg_{\text{int}}(\mathcal{C}) + \deg_{\text{ext}}(\mathcal{C})}, \tag{4}$$

which favors subgraphs with few connections to other parts of the graph. Possible combinations of the above measures are numerous; in this paper we use the product as a cluster quality measure:

$$f(\mathcal{C}) = \delta_\ell(\mathcal{C}) \cdot \delta_r(\mathcal{C}) = \frac{2 \deg_{\text{int}}(\mathcal{C})^2}{|\mathcal{C}|\,(|\mathcal{C}| - 1)(\deg_{\text{int}}(\mathcal{C}) + \deg_{\text{ext}}(\mathcal{C}))}. \tag{5}$$

The complexity of optimizing Equation 5 can be studied through the decision problem of whether a given graph G has a k-vertex subgraph $\mathcal{C}$ for which $f(\mathcal{C}) \geq \gamma$ for some fixed $k \in \mathbf{N}$ and $\gamma \in [0, 1]$. Especially, we are interested to know whether there is such a subgraph that contains a given vertex v. Both $\delta_\ell(\mathcal{C})$ and $\delta_r(\mathcal{C})$ alone correspond to NP-complete decision problems; the complexities of these and other cluster fitness measures are discussed in a separate paper [11].

2.1 Computation by Local Search

Calculation of the proposed fitness measure only requires the adjacency lists of the included vertices. Therefore, a good approximation of the optimal cluster

[1] For $|\mathcal{C}| \in \{0, 1\}$, we set $\delta_\ell(\mathcal{C}) = 0$.

for a given vertex can be obtained by *local search*. To locate a cluster containing a given vertex $v \in V$ from a graph $G = (V, E)$, we stochastically examine subsets of V containing v, and choose the candidate with maximal f as $\mathcal{C}(v)$. The initial cluster $\mathcal{C}'(v)$ of a vertex v contains v itself and all vertices adjacent to v. Each search step may either add a new vertex that is adjacent to an already included vertex, or remove an included vertex. Upon the removal of $u \in \mathcal{C}'(v)$, $u \neq v$, the connected component containing v becomes the next cluster candidate. Redefining $\deg_{\text{ext}} = |\{\langle u, v\rangle \in E \mid u \in \mathcal{C}, v \in V \setminus \mathcal{C}\}|$ allows clustering directed graphs. Our clustering of a 32,148-vertex directed graph representing the Chilean inter-domain link structure is discussed in [12].

The method is well-suited for memory-efficient implementation: if the graph is stored as adjacency lists, $\langle v : w_1, w_2, \ldots, w_{\deg(v)}\rangle$, only one such entry at a time needs to be retrieved from memory. For n vertices, the entries can be organized into a search tree with $\mathcal{O}(\log n)$ access time. The search needs to maintain only the following information:

(a) the list of currently included vertices $\mathcal{C}$,
(b) the current internal degree $\deg_{\text{int}}(\mathcal{C})$ (Equation 1), and
(c) the current external degree $\deg_{\text{ext}}(\mathcal{C})$ (Equation 3).

When a vertex v is considered for addition into the current cluster candidate $\mathcal{C}$, its adjacency list is retrieved and the degree counts for the new candidate $\mathcal{C}' = \mathcal{C} \cup \{v\}$ are calculated as follows:

$$\deg_{\text{int}}(\mathcal{C}') = \deg_{\text{int}}(\mathcal{C}) + k, \qquad \deg_{\text{ext}}(\mathcal{C}') = \deg_{\text{ext}}(\mathcal{C}) - k + \ell, \tag{6}$$

where $k = |\mathcal{C} \cap \Gamma(v)|$ and $\ell = \deg(v) - k$, $\Gamma(v)$ denoting the set of neighbors of vertex v. The removal of vertices from a cluster candidate is done analogously, subtracting from the internal degree the lost connections and adding them to the external degree. The memory consumption is determined by the local structure of the graph. The order of the initial cluster is limited from above by the maximum degree of the graph Δ plus one; in natural graphs, usually $\Delta \ll n$ and $|\mathcal{C}| \ll n$. Hence examining the adjacency lists of the vertices included in the final cluster candidate takes $\mathcal{O}(\Delta \cdot |\mathcal{C}|)$ operations. The extent to which the graph is traversed depends on the local search method applied.

3 Experiments

We have conducted experiments on natural and generated nonuniform random graphs. As natural data, in addition to the web graph discussed in Section 2, we used *collaboration graphs* [7]. We guide the local search with *simulated annealing* [4]. For generalized caveman graphs [13] consisting of a set of interconnected dense subgraphs of varying order, the method correctly identifies any dense "cave" regardless of the starting point; an example is shown in Figure 1. For illustrations of collaboration graph clusterings, see [13]. In other work [14], we also discuss the applicability of the clustering method in mobile ad hoc networks for improved routing.

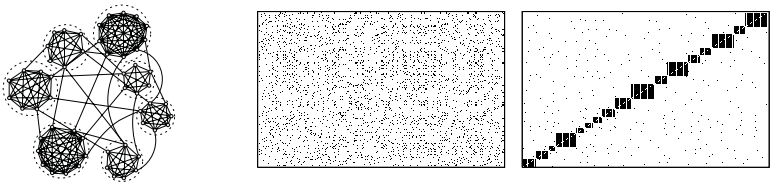


Fig. 1. On the left, an example caveman graph with 55 vertices and 217 edges; each cave (encompassed by a dotted line) is correctly identified as a cluster by the local method. On the right, the adjacency matrix of a caveman graph with 210 vertices and 1,505 edges; the left one uses random vertex order, reflecting very little structure, whereas the one on the right is sorted by the clusters found and reveals the graph structure

As local search procedures are not prohibited from traversing further away in the graph or revisiting parts of the graph, it is interesting to examine whether the extent to which the search traverses the graph has a significant effect on the clusters that the algorithm chooses. We clustered the largest connected component of a scientific collaboration graph with 108,624 vertices and 333,546 edges. We varied the number of independent restarts $R \in \{20, 40, \ldots, 100\}$ per search vertex and the number of cluster modification steps S (from 200 to 1,000 in increments of 100) taken after each restart for simulated annealing. The Figure 2 shows the ratio of the number of vertices visited during the search to the final cluster order, averaged over 100 vertices selected uniformly at random; the final orders are plotted for reference. Figure 2 plots the ratio of the number of vertices visited and the final order of the selected cluster using R restarts of S steps averaged over 50 randomly selected vertices. The extent to which the graph is traversed grows much slower than the number of modification steps taken, implying high locality of the search. As the iteration count is increased, the relative difference gets smaller, which indicates that the number of vertices visited practically stops growing if the increasing possibility for random fluctuations is ignored. The distributions of the cluster orders over three R/S-pairs of the same graph are shown on the right in Figure 2; the distribution changes very little as the parameters are varied, indicating high stability of the method.

We compared the clusterings obtained with the local method to the clusterings of *GMC (Geometric Minimum Spanning Tree Clustering)* with additional linear-time post-processing and *ICC (Iterative Conductance Cutting)* [1] for caveman graphs of different orders. For each graph, we compared the clusters of each vertex obtained with the three methods by calculating what fraction (shown in percentages) of the vertices of a cluster $\mathcal{A}$ determined by one method are also included in the cluster $\mathcal{B}$ determined by another method. Table 1 shows the results for a caveman graph with 1,533 vertices and 50,597 edges; the results

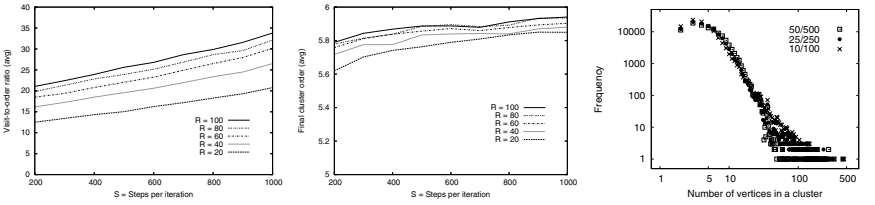
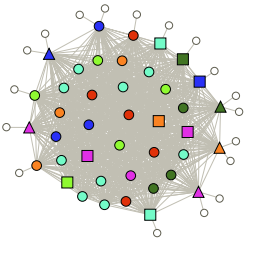


Fig. 2. On the left, the ratio of the number of vertex visited (i.e., the visit count for an (R, S)-pair) to that of the number of vertices selected in the final cluster (i.e., the cluster order) averaged over 100 vertices selected uniformly at random and repeated 50 times per vertex. In the middle, the average final cluster orders of the same experiment set. On the right, the distribution of the number of vertices per cluster for a large collaboration graph (with 108,624 vertices and 333,546 edges) for three different R/S-pairs, where $R \in \{10, 25, 50\}$ and $S = 10R$

Table 1. Denote by $\mathcal{A}$ the cluster chosen by one method for vertex v, and by $\mathcal{B}$ the cluster chosen for v by another method. If the two methods agree, the *overlaps* $a = |\mathcal{A} \cap \mathcal{B}|/|\mathcal{B}|$ and $b = |\mathcal{A} \cap \mathcal{B}|/|\mathcal{A}|$ are high. For three clusterings of a caveman graph, the percentages p of vertices for which the values a and b fall into a certain range are shown. The values are to be interpreted as follows: if $a = a_1$ and $b = b_1$, then a_1 percent of cluster $\mathcal{B}$ (the method of the right column) is included in $\mathcal{A}$ (the method of the left column) and b_1 percent of cluster $\mathcal{A}$ is included in $\mathcal{B}$. The figure on the right shows a single cave in a 649-vertex graph; the small circles are neighbors in other caves. The shape of the vertex indicates its cluster for the post-processed GMC (with three clusters overlapping the cave) and the color indicates the clustering of ICC (seven clusters overlap); the local method selects the entire cave for any start vertex

Local	GMC		Local	ICC		GMC	ICC	
a	b	p	a	b	p	a	b	p
all	all	74	all	$(11, 14)$	45	all	$(5, 14)$	71
all	$(74, 95)$	14	all	$(22, 27)$	12	all	$(22, 34)$	10
all	$(2, 24)$	4	all	$(5, 7)$	36	all	$(40, 55)$	2
$(86, 97)$	all	5	all	$(46, 54)$	6	$(80, 91)$	$(7, 31)$	7
$(3, 57)$	$(5, 87)$	3				$[50, 67)$	$(5, 20]$	4
						$(71, 89)$	$(45, 55)$	3
						$(9, 46)$	$(2, 100]$	3



for the smaller graphs allow the same conclusions. ICC splits the caves into small clusters, which is a sign that it fails to recognize the cave boundary on which the density jump takes place. GMC and the local method agree in a majority of cases exactly in cluster selection, and even when they differ, one is usually a subset of the other. GMC and ICC agree poorly with each other.

4 Conclusions

In this paper we have defined a local method for clustering. The method requires no parameters to be determined by the user, nor does it estimate parameters from the dataset; only the local search method being employed may require parameters. No knowledge of the structure of the graph as a whole is required, and the implementation of the method can be done efficiently. The experiments show that approximate clustering with our measure produces intuitively reasonable clusters without extensive traversing of the graph. Employing a local method in the presented way is likely to produce an approximation of some global clustering method; we hope to determine as further work how our local method relates to other methods, such as spectral clustering discussed in [3]. For another local method, see our recent paper [10].

Acknowledgments

This research was supported by the Academy of Finland under grant 206235 and the Nokia Foundation. We thank Marco Gaertler for providing the GMC and ICC clusterings, Kosti Rytkönen for the assisted use of his graph visualization software, and Pekka Orponen and Christos Faloutsos for their valuable comments on this work.

References

1. U. Brandes, M. Gaertler, and D. Wagner. Experiments on graph clustering algorithms. In G. Di Battista and U. Zwick, eds, *Proc. of ESA*, vol. 2832 of *LNCS*, pp. 568–579, Heidelberg, Germany, 2003. Springer-Verlag.
2. S. Guha, et al. Clustering data streams. In *Proc. of FOCS*, pp. 359–366, Los Alamitos, USA, 2000. IEEE Comp. Soc. Press.
3. R. Kannan, S. Vempala, and A. Vetta. On clusterings — good, bad and spectral. In *Proc. of FOCS*, pp. 367–377, Los Alamitos, USA, 2000. IEEE Comp. Soc. Press.
4. S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
5. J. Kleinberg and S. Lawrence. The structure of the web. *Science*, 294(5548):1849–1850, 2001.
6. M. Mihail, et al. On the semantics of Internet topologies. Tech. report GIT-CC-02-07, Atlanta, USA, 2002.
7. M. E. J. Newman. The structure of scientific collaboration networks. *Proc Natl Acad Sci USA*, 98(2):404–409, 2001.
8. M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, 2004.
9. L. O'Callaghan, et al. Streaming-data algorithms for high-quality clustering. In *Proc. of ICDE*, pp. 685–694, Los Alamitos, USA, 2002. IEEE Comp. Soc. Press.
10. P. Orponen and S. E. Schaeffer. Local clustering of large graphs by approximate Fiedler vectors. In S. Nikoletseas, ed., *Proc. of WEA*, *LNCS*, Heidelberg, Germany, to appear. Springer-Verlag.

11. J. Šíma and S. E. Schaeffer. On the NP-completeness of some cluster fitness measures, 2005. In preparation.
12. S. E. Virtanen. Clustering the Chilean web. In *Proc. of LA-Web*, pp. 229–231, Los Alamitos, USA, 2003. IEEE Comp. Soc. Press.
13. S. E. Virtanen. Properties of nonuniform random graph models. Research Report A77, TCS, TKK, Espoo, Finland, 2003.
14. S. E. Virtanen and P. Nikander. Local clustering for hierarchical ad hoc networks. In *Proc. of WiOpt'04*, pp. 404–405, 2004.