Analysis of Massive Networks

György FRIVOLT*

Slovak University of Technology
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
frivolt@fiit.stuba.sk

Abstract. Massive networks can be observed from various aspects of the world. Nowadays more massive networks have measurable and analyzable form: informational networks such as the WWW, networks of social interactions like call or SMS graphs can be captured. We give an overview of operations, representations and tools for modeling graphs. The intention is to build a basis for further development of a tool for analyzing massive graphs. Finally we introduce some introductory measured properties of a network of SMS communication.

1 Introduction

Those networks which are interesting for research usually perform huge number of vertices and edges. The huge amount of data forces us to think about how to tackle special and computational complexity. Often the data does not fit the system memory. Just for a glance: the crawls of a search engine has to process around 200 million web pages and 2 billion hyperlinks [Newman2003, page 10], the number of communications in a phone network can scale as 50 million for a month.

There are some ongoing project dealing with massive network analysis and visualization (Pajek, JUNG, InFlow, DyNet, Cyram NetMiner, etc.). There are functionalities which are usually aimed by these products.

- *Scalability* - the range of network size the software can cope with
- *Analysis of functionalities* - ranking vertices according to their importance, centrality measurements, clustering

* Supervisor doc. Ing. Bieliková, PhD., Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

M. Bieliková (Ed.), IIT.SRC 2005, April 27, 2005, pp. 35-40.

- *Dynamic network modeling* - analysis of changes in network over time, network evolution prediction functionalities
- *Graph generation* - utilities for graph generation, small-world networks, scale-free graphs
- *Graph visualization* - approaches for visualization of the graph, providing different layouts

Most of the products provide the possibility to export graphs to different formats (GraphML, GraphEd, GML, Graphviz dot, etc.). Naturally for software products the licensing is also an important matter of discussion. There are products both freely available for non-commercial purposes (Pajek) or commercial software (Cyram NetMiner).

Analysis of a software product for network analysis will be described further. An approach for cluster-cutting will be described and measured values of clustering coefficient and average degree of SMS network is served. We conclude the paper with listing further goals.

2 Operations on graphs

2.1 Representations

We distinguish the following types of graph representation (providing different interfaces) useful for different operations:

- *Database storage* – The primary way of representation of the graph is to store it in a well indexed database. The neighbors of any selected vertex should be simply selectable.
- *Vector structure* – For every vertex a list of its neighbor vertices is stored in the memory.
- *Matrix representation* - The adjacency matrix serves much better representation of graph for several operations. The co-citation matrix or importance ranking algorithm PageRank and HITS are defined for operating on adjacency matrixes. However, as most of the real networks are sparse, therefore there is a huge waste of the system memory when bigger portion of a massive graph is represented as matrix. The possibility to store the graph as a list of vectors and operating with it as a matrix would be the best compromise.

2.2 Manipulations

Performing graph manipulation can involve two different type of processing of the graph:

- *Generate-mode* - A new graph is produced, which is completely independent of the originally processed one. Generation of a graph from the source graph causes computational effort when it is executed, but every operation is performed on the produced graph afterwards.
- *Wrap-mode* - The graph manipulation decorates the graph under processing and does not create physically any graph. The produced graph is a kind of a view of the processed graph. This mode causes small effort when it is executed, but less computation when the produced graph is being processed. Also the spatial demands of this operation should be much smaller, as no graph is generated.

2.3 Operations

Graph operations should serve a basis for making decisions for the user or should present a list of results which helps navigation in the network. The following operations were chosen as probably the most relevant for these purposes.

- *Clustering* – Identification of clusters/community structures in the network. The input of the clustering operation is a graph and generates a list of clusters, with vertices they contain, and a hierarchy of clusters.
- *Shrinking* – Shrinking sets of vertices to one vertex, for instance shrinking identified clusters to vertices.
- *Filtering* – Operations for filtering out edges and vertices with given criteria.
- *Co-citation network* – Operation over the adjacency matrix: $E^T E$
- *Ranking* – Implementation of vertex/edge importance ranking, such as centrality measurements [1], PageRank [6] or HITS [2]

Shrinking, clustering and generation of cluster hierarchy are operations of *global decomposition*. *Local decompositions* are cutting out a part of the graph (for instance vertices of a component) or shrinking all but one cluster produces a context of the left alone cluster. Fig. 1 shows an illustration of the decompositions.

Local cluster cutting operation produces a subgraph of the input graph, which is exploited based on an initial seed of vertices.

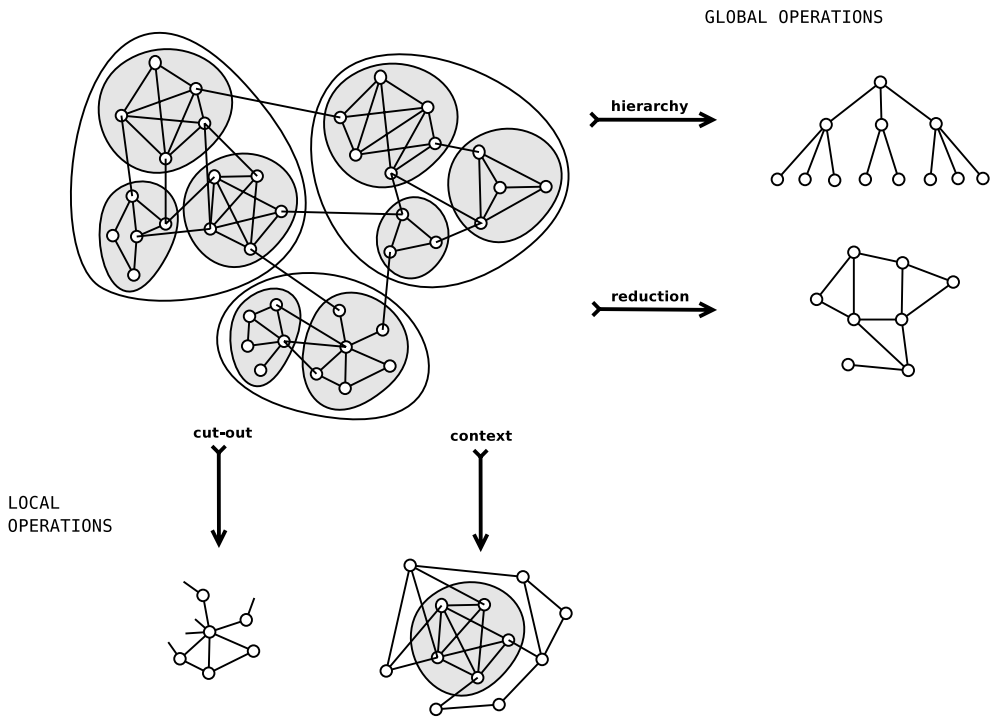

Fig. 1. Graph operations.

2.4 A cluster-cutting approach

Changing the graph database representation to matrix or vector representation may not be always straightforward. The intention of storing the graph in database is keep possibly the all graph on an external storage. For the most of the graphs the network stored externally does not fit the memory, therefore approach for cutting out a portion of the whole data set needs to be proposed.

Our approach is a modified breath-first search algorithm. Whereas the BFS search prioritizes the vertices based on the distance from the seed vertices, we propose to consider priorities also as the rate of the number of neighbors already cut-out to the total number of adjacent vertices.

Requires: graph *G(V,E)*, seed of initcial vertices *S*, maximum number of vertices to cut *L*

Ensures: set of cut-out vertices *C*

$C = S$

$list = S$

while *list* is not empty and $|C|<L$:

  $v = pop(list)$

  add *v* to *C*

  for all neighbours *k* of *v*:

    if *k* in *list*:

      rerank *k* in *list* with priority:

$$\frac{number\_of\_k\_neighours\_in\_C}{total\_number\_of\_k\_neighbors}$$

    else:

      add *k* to *list* with priority:

$$\frac{1}{total\_number\_of\_k\_neighbors}$$

return *C*

Alg. 1. Finding cut-out vertices.

3 Analysis of SMS communication network

A simple strategy to cope the huge amount of data was used by us to get some properties of the network of SMS communication. The network contained over 15 million edges/SMS communications, only the evidence of one communication was captured in the dataset, therefore the real number of SMS messages could be much higher. The analyzed graph was undirected, any communication from actor a to b was considered as edge $\{a,b\}$ without orientation. The communication was captured over one month. The identification of the communication actors were coded, therefore no personal rights could be disused.

Our primary intention was to touch the data set to get the first impression about its processing. We used a *mysql* database table with two columns as data storage. The two columns coded the initiator and the target phone number. Although the network could be taken as directed, we did not take any advantage of the orientation of the edges. The data processing was implemented in *python*.

We measured the clusterization coefficient [WattsStrogatz98], average degree. Randomly chosen vertices were analyzed against the target network properties.

The computation was executed on computer with processor Intel Centrino 1.4GHz, 512Mb RAM, sufficient HDD space, with Linux OS. The clusterization coefficient have been found 0.131 for the SMS communication network, which is significantly higher as it would be expected for random graphs. This property of the graph means, that in general more than 13% of those people who we send SMS

messages also send messages among each other. The average number of "SMS neighbors", the average degree of the graph was found to be 6. The computation took 6 hours, 37 minutes. This time will have to be reduced in the future by processing the data set in the memory.

4 Conclusions

We presented a short description of functionalities required by tool for massive graph analysis. An algorithm for cutting out a part of the graph was proposed. This algorithm is necessary for finding a portion of the graph based on the given initial seed, which can be further processed in the system memory. Finally some first results of the graph were presented.

The proposed algorithm for graph cutting has to be tested on real data set. There are also other properties of the graph which are worth of measuring: degree distribution, degree-correlation [3]. Our main goal is to identify the network clusters and rank the importance of network vertices and edges.

Acknowledgement: This work was supported by Science and Technology Assistance Agency under the contract No. APVT-20-007104.

References

1. Brandes U.: A Faster Algorithm for Betweenness Centrality. *Journal of Mathematical Sociology*, Vol. 25 (2001), 163-177.
2. Kleinberg J.M.: Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, Vol. 46 (1999), 604-632,
3. Newman, M.E.J.: Assortative mixing in networks. *Int. Physical Review Letters*, Vol. 89, No. 20 (2002), cond-mat/208701
4. Newman, M.E.J.: The structure and function of complex networks. *Int. Physical Review Letters* (2003), cond-mat/0303516
5. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. *Int. Nature*, Vol. 393, (1998), 440-442.
6. Page L., Brin S., Motwani, R., Winograd T.: The PageRank Citation Ranking: Bringing Order to the Web, *Standford Digital Libraries, Technologies Project* (1998), unpublished