Graph Clustering Algorithms

Aleksandr Salo, Rovshen Nazarov

***Abstract*—In this article we provide a brief overview of the few existing graph clustering algorithms that are commonly used in bioinformatics. Then we evaluate their performance and accuracy based on the sample protein-protein interaction network as an input and subgraphs representing potential protein complexes as an output. In particular we use f-measure to evaluate the clustering results by comparing to protein complex data provided. For measuring the accuracy of the algorithms, we compute an f-score for each output cluster by selecting the maximum f-score to a protein complex, and average the f-scores of all output clusters. After evaluation we elaborate our own algorithm that tries to outperform the existing ones.**

***Index Terms*—datamining, graphs, algorithms, bioinformatics**

I. Introduction

As a basis for the future comparison we implemented three different graph clustering algorithms:

1) Common Neighbors Cut
2) Common Neighbors Merge
3) Seed Growth: Graph Entropy

Text file describing all the edges of the graph was used as an input. Basic characteristics of the graph are following:

1) Unweighted
2) Disconnected
3) Undirected
4) Size ($|V|$) = 2526
5) Edges ($|E|$) = 11450
6) Density = 0.00179

Let us know highlight the main ideas of implemented algorithms.

A. Common Neighbors Cut

This top-down hierarchical method generates non-overlapping clusters recursively dividing the initial graph using dissimilarity measure. The main idea is: ”Less common neighbors two vertices share, more dissimilar they are.” [1]

Algorithm [3]:

1) Iteratively eliminate the edge between the most dissimilar vertices based on a similarity function, until the graph is separated
2) Recursively apply (1) into each subgraph
3) Repeat (1) and (2) until all subgraphs reach a density threshold

The main time consuming operation in the algorithm that repeatedly occurs is defining whether the graph is still connected (after removing the edge between two most dissimilar vertices) or not. Our implementation uses depth-first-search algorithm (which is very efficient) for deciding this question.

B. Common Neighbors Merge

This algorithm is a mirrored brother of the previous one. Instead of recursively dividing the graph we initiate each node of the graph as a separate cluster. Then we merge most similar nodes until threshold value is violated. So the main idea is: ”More common neighbors two vertices share, more similar they are.” [2]

Algorithm [3]:

1) Find the most similar vertices from different clusters based on a similarity function
2) Merge the two clusters if the merged cluster reaches a density threshold
3) Repeat (1) and (2) until no more clusters can be merged

In order to avoid O(n^4) calculation of the most similar nodes among all the clusters we initially calculate the table of distances for each pair of nodes and then look for the most relevant values.

C. Seed Growth: Graph Entropy

This algorithm is a typical density-based algorithm which produces overlapping clusters via seed growth. The main idea is: ”Search for local optimization using a modularity (density) function.” [4]

Algorithm [3]:

1) Select a seed node among free vertices, and include all neighbors of the seed node into a seed cluster
2) Iteratively remove a neighbor if removal decreases graph entropy
3) Iteratively add a node on the outer boundary of a current cluster if addition decreases graph entropy
4) Output the cluster with the minimal graph entropy
5) Repeat (1), (2), (3), and (4) until no seed node remains

Here entropy is a density measure (computable in O($|V|$) time) based on probabilities of having inner or outer links. Seed selection is based on the vertex degree: the more degree is the more chances that the vertex would be a good seed.

Clearly, this algorithm runs very fast since it doesn’t use recursion or nested loops among all the vertices.

D. Implementation notes

For all our algorithms we used the arbitrary chosen threshold value of 0.3. Also we tried to treat algorithms the same way and implement them using the same data structures (however authors of original papers used slightly different implementation).

Our estimation is that the following tests should reflect the real measure of performance and accuracy.

Cho, Young-Rae is with the Department of Electrical and Computer Engineering, Baylor University, Waco, TX, 76706 USA

II. Performance and Accuracy Comparison

For evaluation of accuracy we used the f-measure to evaluate the clustering results by comparing to protein complex data (ground truth), where f-measure is a harmonic mean of *recall* and *precision*:

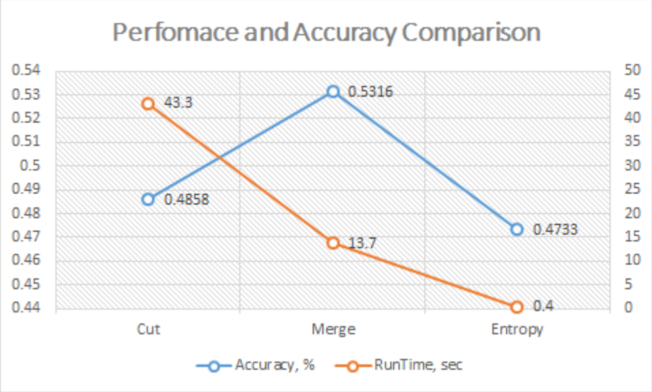
$$\text{f-measure} = \frac{2*recall*precision}{recall+precision},$$

Recall (Sensitivity or True positive rate) = $\frac{|X \cap Y|}{|Y|}$
Precision (Positive predictive value) = $\frac{|X \cap Y|}{|X|}$

Computing f-measure for the algorithm's accuracy is a three step process:
1. Compute all the f-scores for each resulting cluster.
2. Assign maximum f-score to each cluster.
3. Take average f-score as an f-measure of the algorithm's accuracy.

Hereby accuracy could vary from 100% (each of the resulting clusters are in the ground truth clusters) to 0% (none of them fit).

Runtime is simply an execution time in seconds.

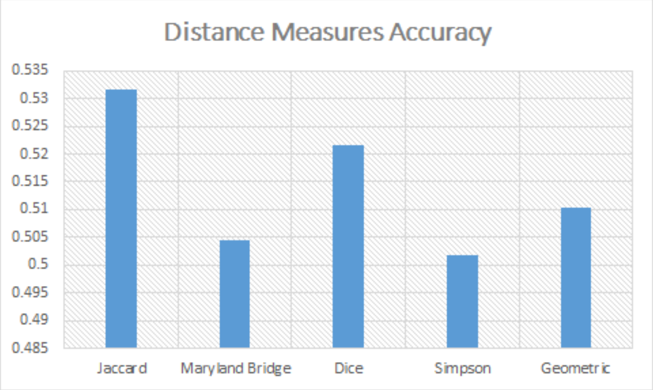


Clearly, graph entropy algorithm is by far the fastest. However common neighbors merge perform slightly better results yet more efficient than common neighbors cut.

A. Distance measure

While common neighbors merge algorithm produces good performance the accuracy could be varied with the use of different distant measures. Below we plot the accuracy for each of the following distance measures:

- Jaccard Coefficient $S(x,y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}$
- Maryland Bridge $S(x,y) = \frac{1}{2}(\frac{|N(x) \cap N(y)|}{|N(x)|} + \frac{|N(x) \cap N(y)|}{|N(y)|})$
- Dice $S(x,y) = \frac{2|N(x) \cap N(y)|}{|N(x)| + |N(y)|}$
- Simpson $S(x,y) = \frac{|N(x) \cap N(y)|}{min(|N(x)|, |N(y)|)}$
- Geometric $S(x,y) = \frac{|N(x) \cap N(y)|^2}{|N(x)||N(y)|}$



III. Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Appendix A
Proof of the First Zonklar Equation

Some text for the appendix.

Acknowledgment

The Σ authors would like to thank...

References

[1] Radicchi, Filippo, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and Identifying Communities in Networks. Proceedings of the National Academy of Sciences of the United States of America 101, no. 9 (March 2, 2004): 265863. doi:10.1073/pnas.0400054101.

[2] Brun, Christine, Carl Herrmann, and Alain Gunoche. Clustering Proteins from Interaction Networks for the Prediction of Cellular Functions. BMC Bioinformatics 5, no. 1 (July 13, 2004): 95. doi:10.1186/1471-2105-5-95.

[3] Cho, Young-Rae. ”CSI 4352, Introduction to Data Mining.” 10/16/2014 (n.d.): n. pag. Web.

[4] Chiam, Tak C., and Young-Rae Cho. Accuracy Improvement in Protein Complex Prediction from Protein Interaction Networks by Refining Cluster Overlaps. Proteome Science 10, no. Suppl 1 (June 21, 2012): S3. doi:10.1186/1477-5956-10-S1-S3.

John Doe Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.