Amartya Chakraborty
Rensselaer Polytechnic Institute
Dr. Mohammed Zaki

Studying Community Detection Algorithms on Data Science Citation Networks

Introduction:

In academia, citations are a vital part of any publication. They can add background to the topic, results from previous experiments and studies done in the field and overall valuable knowledge to help guide current research. Academic publications are present in almost every field or area whether it is in Computer Science or Chemistry. One area which has been looked into is the analysis of citation networks where a graph is based upon the nodes which would be the papers and edges that would be between papers which cite one another. In this paper, we utilized the Microsoft Academic Graph database [11, 15] to construct a graph of papers related to Data Science. From there, we ran six different clustering algorithms and then evaluated the performance of the algorithm on the final clusters created.

Related Work

There has been prior work on trying to cluster academic publications. One paper which was consulted as part of this project had worked on clustering algorithms on scientific publications [14]. In this case, we are focusing on Data Science publications but the idea remains the same. The paper discussed how Infomap was able to give some of the top results as part of that study. Hence, that clustering algorithm was utilized as part of this study. In addition, this study also utilized the idea presented by a previous study on trying to link academia papers to one another based on citations [16]. That study contributed the idea of the one-hop expansion [16] which was utilized for this project in creating the citation graph which is studied here.

Method:

The dataset used for this experiment was from the Microsoft Academic Graph. This contained a total of 166,192,182 papers. In this dataset, each paper is represented as a JSON object. Python was used to parse the object into a readable format and then the paper id, title, abstract and references were all extracted. For this particular paper, the graph to be constructed involved papers which had the terms “Data Mining”, “Machine Learning”, “Artificial Intelligence”, and “Data Science” in the title or abstract of the paper. If the paper contained any of these terms as part of its title or abstract, it was added to a group of papers. For these four terms, there were a total of 112,000 papers that were part of the initial group. The next step involved a one hop expansion on this initial group of papers. The way this worked is if a paper is not part of the initial group and either it includes a paper from the initial group as part of its references list or it is a part of a paper’s references list in the initial group, then it will be added to the graph. To maintain an appropriate size, only a one hop expansion was done. The graph consisted of having each paper in the final group as its own node and edges were between papers in this group that were either referenced by or referenced each other. The final graph consisted of 1983837 nodes and 29541428 edges between them. Additionally, the graph is undirected. Python scripts were utilized to obtain this graph. From here, six different clustering algorithms were

utilized to cluster this graph. The algorithms were MCODE[1], Louvain [2], Graclus[3-5], Infomap [7,10], IPCA [8], and Label Propagation [9].

In the case of Graclus [6] and Infomap [7], the actual code from the respective project websites were utilized. Meanwhile, for the Louvain method and Label Propagation algorithms, an implementation by Dr. George Slota of Rensselaer Polytechnic Institute [13] was utilized to obtain the appropriate clusters. Dr. Slota developed this implementation as part of one of the courses he teaches at the university. The implementations of MCODE and IPCA were obtained from a Github repository [17]. The Github repository focuses on clustering protein structures but the reason that these two algorithms were selected to be tested is because as is the case with proteins relationships with one another, different papers also have connections to other papers. Hence, this was done out of curiosity to utilize these two clustering algorithms to observe how they clustered this graph.

Each of these clustering algorithms were then run on the same graph and the results were then analyzed through a number of different methods. The conductance and modularity were calculated for each of the clustering algorithms and used as an evaluation metric of the cluster assignment. Both of these values were percentages. The conductance statistic measured how easily two papers in different clusters can access each other based on edge traversal. A lower conductance value which ideally would be close to 0. Meanwhile, modularity is looking at how well the graph is split into different clusters. Thus, a value closer to 1 would be ideal in this case.

In addition, the clusters were analyzed by computing the average cosine similarity between the combination of a paper's title and abstract within the other papers that are part of the same cluster. This is the intra-cluster cosine similarity metric which was obtained. There was one element of this experiment that was impacted by time constraints. The goal would be for each cluster, to compute the cosine-similarity of the TF-IDFs between each paper in that cluster with all of the other papers in the cluster. Thus, with a cluster of size N, there would be a total of (N * (N – 1)) / 2 comparisons made. This computation would be done through the Scikit-learn library [12] in Python. However, when dealing with clusters of extremely long size, this process simply could not be completed in a reasonable amount of time. Therefore, one restriction was set that for any cluster with more than 300 papers, the top five papers in that cluster with the highest overall degree were selected. From there, the intra-cluster cosine similarity was computed just with those five papers among themselves first and the other papers in that cluster. This significantly reduced the amount of cosine similarity computations done and allowed for the program to finish running in a reasonable time frame. The paper degree was measured by how many other papers it either references or is referenced by in the specific Data Science graph we are studying.

In addition, the inter-cluster cosine similarity between the papers in a particular cluster with all of the papers from different clusters was also studied. The idea of this evaluation was to calculate the average cosine similarity each paper in a cluster with the papers from another cluster. This would help determine if any two clusters may have papers with more similar titles and abstracts than any other clusters from the graph. Like the intra-clustering analysis, one constraint was placed on this analysis as well. Only clusters with at least 25 papers would be studied and analyzed since clusters such as MCODE and IPCA had many trivial clusters or very small clusters. These small clusters would not bring about the same insight in terms of similarities within the papers from different clusters. Once this filter has been applied, from each cluster, the top 10 papers with the highest total degree would be taken and the average cosine similarity would be calculated with the 10 highest degree papers from the other clusters. The

overall expectation is that that the inter-cluster cosine similarity would yield a smaller value than the intra-cluster cosine similarity values.

One thing to note is the Graclus algorithm took in as input the number of clusters to separate the graph into. The other five algorithms all just continued to cluster papers until there were no more papers remaining which were not part of any cluster.

Experiments were run in a Linux environment with 12 cores and 256 GB RAM. In addition, the code for this project can be found at this link: https://github.com/AmartC/CitationNetwork.

Results:

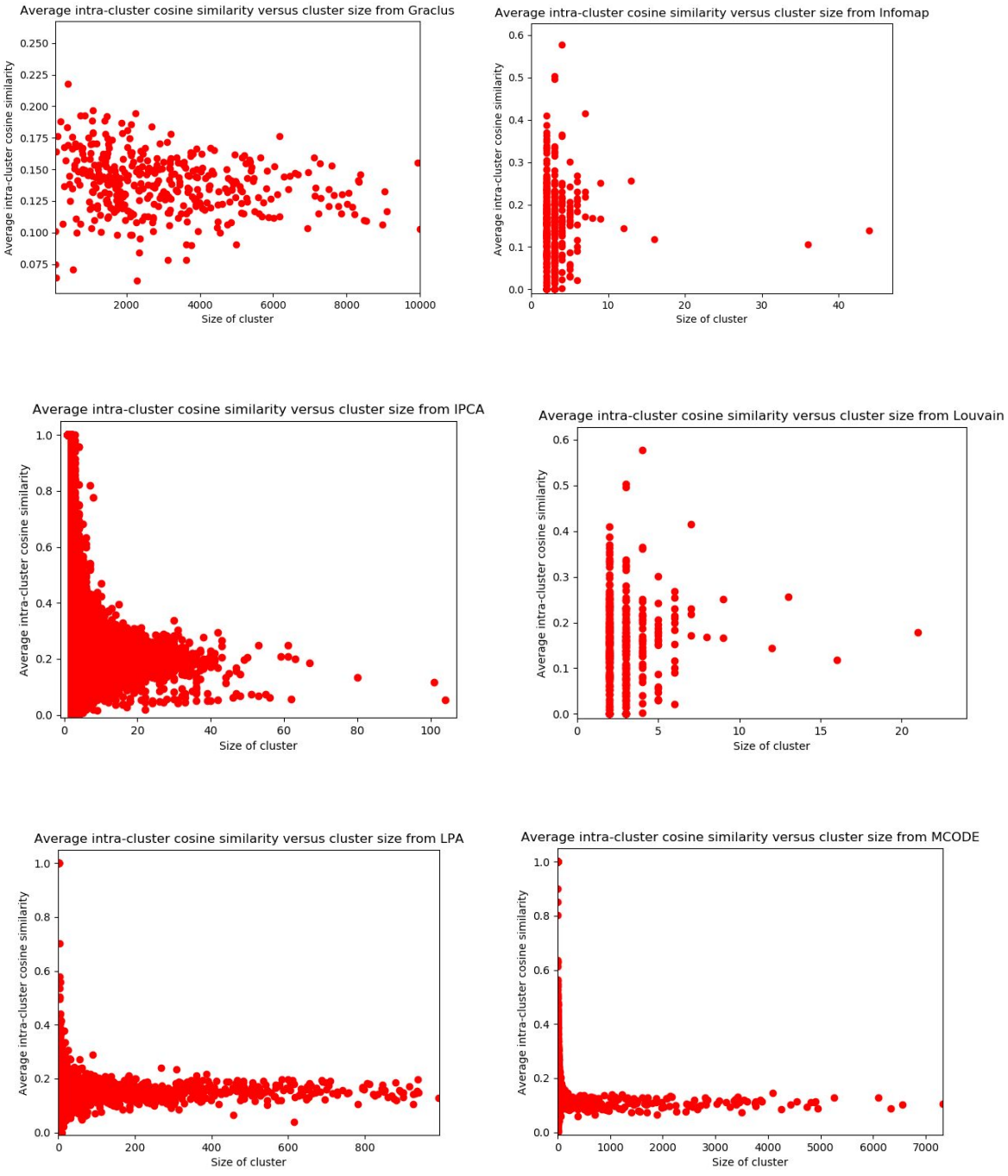
Clustering Algorithm	Number of Clusters	Conductance	Modularity
Graclus	2500	0.422	0.504
Graclus	420	0.380	0.566
Infomap	426	0.009	0.816
IPCA	1,245,129	0.998	0.020
Label Propagation Algorithm	5240	0.364	0.662
Louvain Modularity	415	0.003	0.805
MCODE	1,544,498	0.998	0.024

Table 1: Conductance and Modularity Calculated for Clustering Algorithms

When it comes to conductance, the algorithms with the lowest values were Louvain Modularity and Infomap with scores of 0.003381 and 0.009578 respectively. When observing the table, these two algorithms had two of the three smallest total number of clusters. Graclus [Table 1] had 420 clusters which was the second smallest and its conductance and modularity scores are moderate but not as high as Louvain Modularity and Infomap. As conductance measures how well nodes from different clusters can access one another based on edge traversal, based on this dataset, it would seem to support the idea that using a clustering algorithm to split a graph into a smaller number of clusters may be the best approach to use. This would also support the study which looked into the best clustering algorithms to use on a paper consisting of scientific publications [14] which also discussed in its results how it found Infomap to be one of the better clustering algorithms it looked into. Moreover, both Louvain Modularity and Infomap also had the two highest modularity scores which indicate that these two algorithms had the best methodology in splitting the graph into different clusters.

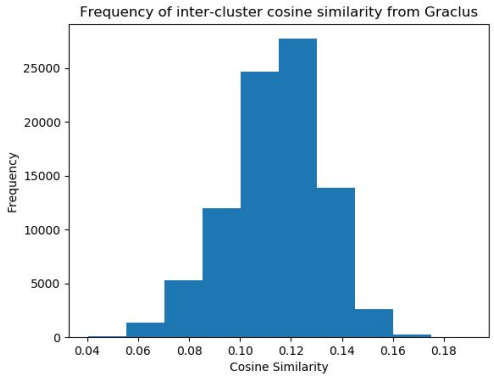
Conversely, the two algorithms which gave the worse results on conductance and modularity were MCODE and IPCA. Both of these algorithms yielded over 1,000,000 clusters and consequently, there were many trivial clusters present. The conductance score of both these methods was 0.998 meaning that nodes from different clusters can visit one another with

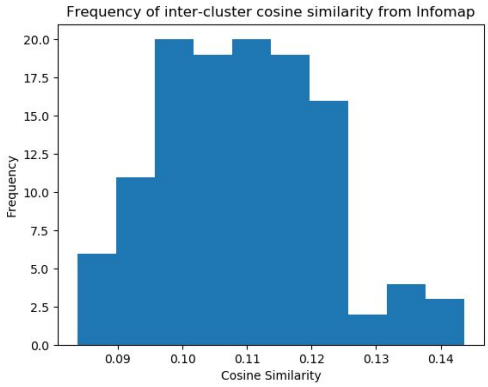
relatively high frequency which is not a desirable feature of a clustering algorithm. In addition, the modularity scores of IPCA and MCODE were very low at 0.020 and 0.024 respectively. This is another indication of relatively poor performance since a low modularity score indicates that the graph cannot be clustered into components very well.

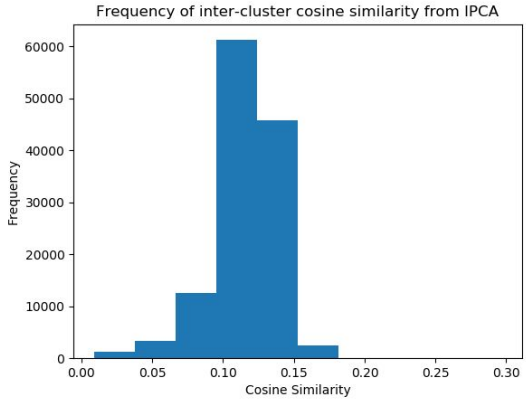


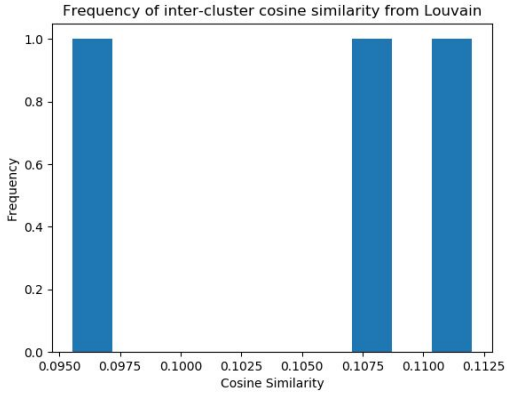
Figures 1-6: Average intra-cluster similarity versus cluster size for Graclus, Infomap, IPCA, Louvain, LPA and MCODE algorithms.

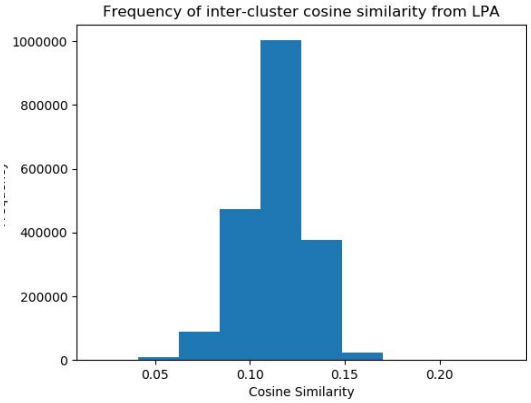
Overall, from the intra-cluster cosine similarity scatter plots which were done with respect to the size of the cluster, no real conclusion can be made with regards if any of the algorithms gave the best results. One finding which was consistent though was that among smaller clusters, higher average cosine similarity values were obtained from the combination of the title and abstract of papers within the same cluster. For any case where there was a similarity of 1, this was likely indicative of a paper being a part of a trivial cluster which means it was the only paper in that cluster. In that case, the average cosine similarity of any trivial cluster is set to 1.0. In general though, as the size of a cluster increased, the average intra-cluster cosine similarity decreased as well, though it also appeared to level off at approximately 0.15. The Graclus [Figure 1], IPCA [Figure 3], Label Propagation [Figure 5] and MCODE [Figure 6] charts both display an inverse relationship between the cluster size and average intra-cluster cosine similarity. As the cluster size increases, the average intra-cluster cosine similarity decreased. Label Propagation. Meanwhile, the charts for Infomap [Figure 2] and Louvain [Figure 4] are a bit different. Both of these charts show that one cluster size has many different average intra-cluster cosine similarity values. This would indicate that cluster size does not have a relationship with the average intra-cluster cosine similarity, which is a different finding from the other four clustering algorithms. However, within those same figures, as the cluster size increased, the average cosine similarity values fluctuate less and begin decreasing. Additionally, the largest non-trivial intra-cluster cosine similarity value was approximately 0.90 which was observed from the MCODE results [Figure 6]. Infomap [Figure 2] and Louvain [Figure 4] which yielded the best results for conductance and modularity had a high of approximately 0.60 for the average intra-cluster cosine similarity value for a non-trivial cluster. For Graclus, the figure pertains to the clustering which led to 420 clusters as that method provided a stronger modularity and conductance score than the method which led to 2500 clusters.

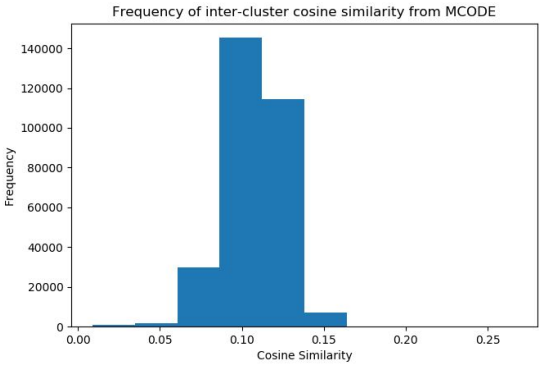

Frequency of inter-cluster cosine similarity from Graclus
Frequency
Cosine Similarity


Frequency of inter-cluster cosine similarity from Infomap
Frequency
Cosine Similarity


Frequency of inter-cluster cosine similarity from IPCA
Frequency
Cosine Similarity


Frequency of inter-cluster cosine similarity from Louvain
Frequency
Cosine Similarity


Frequency of inter-cluster cosine similarity from LPA
Cosine Similarity


Frequency of inter-cluster cosine similarity from MCODE
Frequency
Cosine Similarity

Figures 7-12: Frequency of inter-cluster cosine similarity values from Graclus, Infomap, IPCA, Louvain, LPA and MCODE algorithms respectively.

Overall, the inter-cluster cosine similarity values among the six clustering algorithms were all very close to one another. The most frequent inter-cluster cosine similarity values were between 0.10 to 0.15 for all six algorithms. One thing to note is for the chart of the Louvain algorithm. Only clusters with size of at least 25 papers were studied in this case to avoid the cases of trivial or very small clusters that could cause outlier problems. From there, the average inter-cluster cosine similarity of the top 10 papers from each cluster which had the highest overall degree based on references Thus, in the case of the Louvain algorithm, only three such clusters are present which is why it's chart displays just three bars in total. When comparing these results with the average intra-cluster cosine similarity, the intra-cluster cosine similarity values is overall higher than the most frequent average inter-cluster cosine similarity values from the six clustering methods. However, this can only be seen when the cluster size is relatively small as an inverse relationship was seen between cluster size and the average intra-cluster cosine similarity values. This is an expected result since papers which are part of the same clusters should have more similar title and abstracts than papers from different clusters. One thing to note is for Graclus, the figure pertains to the clustering which led to 420 clusters as that result provided stronger conductance and modularity scores than the method which led to 2500 clusters.

Analysis:

When studying conductance and modularity of each of these clustering algorithms, an inverse relationship with low conductance and high modularity was the goal. The two algorithms which delivered the best results on those conditions were Infomap and Louvain modularity. The conductance for Infomap was 0.009 and the modularity was 0.816. Meanwhile, the conductance for Louvain modularity was 0.003 and the modularity was 0.805. In addition, these two algorithms also had the two of the three smallest total number of clusters which would consequently mean that their average cluster sizes would be among the two highest values. The Graclus algorithm which yielded 420 clusters and had a conductance of 0.380 and a modularity of 0.566 yielded solid results as well compared to the other three algorithms with a much larger number of clusters yielded. This would explain the finding since by splitting up the graph into a smaller number of clusters, there would be a lot more possible papers in a cluster that have a higher similarity to one another based on the contents of the title and abstract. While there is not a clear relationship depicted in terms of smaller numbers of clusters yielding enhanced modularity and conductance scores, a correlation definitely seems to be supported here in terms of aiming to utilize algorithms that can minimize the total number of clusters created. This would rule out algorithms that have a large number of trivial or non-trivial clusters which are extremely small such as MCODE and IPCA. In addition, the average intra-cluster cosine similarity was higher than the average inter-cluster cosine similarity for all six algorithms which was expected. All of the clustering algorithms showed an inverse relationship between the cluster size and average intra-cluster cosine similarity values. Based upon observation, the highest intra-cluster cosine similarity value for a non-trivial cluster was approximately 0.90 as was seen from MCODE [Figure 6]. Label propagation had a high value of approximately 0.70 for a non-trivial

cluster and Infomap and Louvain modularity both had a high value of approximately 0.60 for a non trivial cluster. Thus, what this indicates is that for small sized clusters, MCODE would yield the best results if trying to maximize the largest intra-cluster cosine similarity. However, when it comes to larger clusters, all six algorithms gave approximately the same results. Hence, for larger clusters, the conductance and modularity metrics are a better evaluation tool and thus Infomap and Louvain modularity give the best results.

Future Work:

This project largely studied the clustering algorithm methods done with papers that are based on Data Science topics. Specifically, the constructor of the graph searched for four specific keywords as part of each paper's title and abstract as discussed earlier. These terms "Data Mining", "Machine Learning", "Data Science", and "Artificial Intelligence" are just some of the many different terms that are keywords in the Data Science field. Thus, some expansion to this project may involve trying to construct an even larger graph based on expanding the list of keywords. Another area to look into is how these clustering algorithms perform on other areas of publications. For example, instead of Data Science, if other areas in Computer Science are explored such as Algorithms, Robotics, or Cloud Computing, this clustering mechanism could then be evaluated on its versatility on other areas of Computer Science. In addition, another area to look into is trying to test the clustering algorithm on different fields altogether. This may include Astronomy, Mathematics, Economics among any other research areas. Regardless, future work on this project can be done in a number of different areas.

Conclusion:

Citation networks are a vital component for modern day research. Additionally, clustering a citation network can be an effective way of grouping together papers of similar topics and genres. For the purpose of this experiment, a citation network was constructed from the Microsoft Academic Graph database, only based upon papers which had the terms "Data Science", "Machine Learning", "Data Mining", and "Artificial Intelligence". When looking into the conductance and modularity scores, the two algorithms which gave the best results were Infomap and Louvain Modularity. For very small cluster sizes, MCODE gave the highest average intra-cluster cosine similarity result at approximately 0.90. However, as the cluster size increases, all six algorithms leveled off at approximately 0.15. Additionally, the average intra-cluster cosine similarity was mostly larger than the average inter-cluster cosine similarity. This is understandable because the papers from the same cluster should have more closely related titles and abstracts than papers from different clusters. Thus, as the average intra-cluster cosine similarity values all eventually leveled off as the cluster size increased for all six algorithms, the best metric to look at then is the conductance and modularity results. Hence, the two best algorithms to utilize for clustering citation network graphs based on Data Science topics would be Infomap and Louvain modularity. In addition, it would also be beneficial to split the graph into a smaller number of clusters such as to maximize the modularity and conductance.

Acknowledgement:

A big acknowledgement goes to my Masters project advisor, Dr. Mohammad Zaki of Rensselaer Polytechnic Institute. I have learned so much under his guidance. In addition, another

acknowledgement goes to Dr. George Slota of Rensselaer Polytechnic Institute who taught me a lot about Graph theory and analysis through his courses Graph Theory and Parallel Graph Analysis.

References:

1. Gary Bader, CW Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics.* January 13, 2003.

2. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment.* 2008 (10), P1000. Louvain

3. I. Dhillon, Y. Guan, and B, Kulis. Weighted Graph Cuts without Eigenvectors: A Multilevel Approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 29:11, pages 1944-1957. November 2007. Graclus

4. I. Dhillon, Y. Guan, and B, Kulis. A Fast Kernel-based Multilevel Algorithm for Graph Clustering. *Proceedings of The 11th ACM SIGKDD*, Chicago, IL. August 21-24, 2005. Graclus

5. I. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, Spectral Clustering and Normalized Cuts. *Proceedings of The 10th ACM SIGKDD*, Seattle, WA. August 22-25, 2004. Graclus

6. Graclus Software. The University of Texas at Austin Computer Science.

7. Infomap. http://www.mapequation.org/code.html

8. Min Li, Jianer Chen, Jianxin Wang, Bin Hu, Gang Chen. Modifying the DPClus Algorithm for Identifying Protein Complexes Based on New Topological Structures. BMC bioinformatics. 9. 398. 10.1186/1471-2105-9-398. 2008. Ipca

9. Usha Nandini Raghavan, Reka Albert, Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *The Pennsylvania State University,* University Park, PA. September 19, 2007. Label Propagation

10. Martin Rosvall, Daniel Axelsson and Carl T. Bergstrom. The Map Equation. *European Physical Journal*. 2009. Infomap

11. Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. *In Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*. ACM, New York, NY, USA, 243-246. 2015.

12. Scikit-learn. http://scikit-learn.org/stable/

13. George Slota. Parallel Graph Analysis. Rensselaer Polytechnic Institute. 2017

14. Lovro Subelj, Nees Jan van Eck, Ludo Waltman. Clustering Scientific Publications Based on Citation Relations: A Systematic Comparison of Different Methods. *National Center For Biotechnology Information.* 28 April 2016.

15. Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. ArnetMiner: Extraction and Mining of Academic Social Networks. In Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2008). Pp.990-998. 2008.

16. Luam C. Totti, Prasenjit Mitra, Mourad Ouzzani and Mohammed J. Zaki. A Query-oriented Approach for Relevance in Citation Networks. In Proceedings of the 25th International Conference Companion on World Wide Web: 3rd WWW Workshop on Big Scholarly Data: Towards the Web of Scholars.

17. TruePrice. Python-graph-clustering. https://github.com/trueprice/python-graph-clustering

18. Mohammed J. Zaki. Rensselaer Polytechnic Institute.