Distributed Community Prediction for Social Graphs Based on Louvain Algorithm

Christos Makris, Dionisios Pettas, Georgios Pispirigos

Distributed Community Prediction for Social Graphs based on Louvain algorithm

Christos Makris[1] and Dionisios Pettas[2] and Georgios Pispirigos[3]

[1] Department of Computer Engineering and Informatics, University of Patras, Patras, Greece
makri@ceid.upatras.gr
[2] Department of Computer Engineering and Informatics, University of Patras, Patras, Greece
petta@ceid.upatras.gr
[2] Department of Computer Engineering and Informatics, University of Patras, Patras, Greece
pispirig@ceid.upatras.gr

Abstract. Nowadays, the problem of community detection has become more and more challenging. With application in a wide range of fields such as sociology, digital marketing, bio-informatics, chemical engineering and computer science, the need for scalable and efficient solutions is strongly underlined. Especially, in the rapidly developed and widespread area of social media where the size of the corresponding networks exceeds the hundreds of millions of vertices in the average case. However, the standard sequential algorithms applications have practically proven not only infeasible but also terribly unscalable due to the excessive computation demands and the overdone resources prerequisites. Therefore, the introduction of compatible distributed machine learning solutions seems the most promising option to tackle this NP-hard class problem. The purpose of this work is to propose a novel distributed community detection methodology, based on the supervised community prediction concept that is extremely scalable, remarkably efficient and circumvent the intrinsic adversities of classic community detection approaches.

Keywords: Community Detection, Machine Learning, Distributed Computing, Supervised Learning, Louvain algorithm.

1 Introduction

In recent years, the amount of available information continuously grows, making the need for efficient information representation and data compactness more and more crucial. Therefore, the adoption of data structures such as graphs have frequently been the case, in order to conveniently enhance and readily model the primary information with its corresponding relationship structure. Characteristically, many sectors such as biology, chemistry, sociology, marketing and computer science have embraced information networks since they seem extremely suitable for hierarchical data representation [1]. Especially, in targeted marketing and personalized recommendations, where the concept of retrieving the underlying hierarchical structure from complex networks have

been a major concern. Hence, community detection has become one of the most significant and concurrently challenging problems in graph analysis, having as principal goal the identification of subgroups that contain highly similar entities by only using topological information. The entities similarity is calculated using specific graph topology criteria such as the edge/node connectivity degree, the clustering coefficient, the edge betweenness centrality etc. [6]. The returned subgroups, also known as communities, should be typically densely intra-connected and sparsely inter-connected with each other.

Focusing on the social media, their information can be exceptionally modeled using graphs, where each node could represent a unique user and the connection between two users could generally express any kind of social interaction. Due to the natural human tendency to get associated and mainly interact with peers of similar interests, the formation of virtual clusters and communities can be considered as a doubtless consequence [1] [20]. Therefore, the community detection in social networks can be interpreted as the identification of groups of social media users that are either directly or indirectly connected and tend to interact more often with each other comparing to users of different communities [5]. The concept of finding like-minded user subgroups has indeed proven beneficial in e-commerce, social media marketing and recommendation services.

Profuse measures have been proposed for the retrieved community structure evaluation [1] [20]. Among all, the most broadly used is the modularity which is basically a function that quantifies the average connectivity degree of a given community [1]. Intuitively, modularity seeks to reflect the concentration of remaining edges within communities, after community detection application, compared to a random distribution of connections on the original graph [2]. The modularity values lie between -1 and 1, where positive modularity values reveal a well-identified / well-assessed community structure and vice versa the negative ones indicate a much more complicated subjacent community structure that have not been fully identified yet.

Apart from being an indispensable clustering evaluation benchmark, modularity also serves as the elementary method for graph clustering optimization in the great majority of existing methods and algorithms [3] [7] [10] [12] [13] [18]. Nonetheless, despite its broad use, modularity optimization is inherently classified as a NP-complete problem [1] with typical implementations of high polynomial computational complexity.

Apart from the undisputed modularity maximization computational difficulties, the actual size of real-world social networks acts as another major inhibitory factor. It is a common fact that today's social networks have outreached the billions of user profiles, with Facebook reporting more than 2 billion users in 2018 while the corresponding number for Instagram exceeded the 800 million users. Consequently, for social graphs of such extend, it has been obviously infeasible to even traverse through the entire network using the classic methodologies.

Hence, it seems utterly pointless to even try to apply the standard graph clustering approaches described in [1] [2] [4] [5] [18] since the required calculations outstrip the linear complexity. Thereupon, only scalable distributed solutions that make use of the topological properties retrieved from a small but representative part of the original

graph, could rise to the occasions. As a result, considering that link prediction techniques could be very prolific for tremendously big information networks [8] [9], the community detection problem can alternatively be interpreted as the probability calculation problem of nodes belonging on the same community for each directly connected pair of nodes.
Inspired by the aforementioned idea, a novel distributed methodology, which follows the idea of link prediction, is proposed. Given that Louvain algorithm's [6] performance is generally considered as the golden standard in graph clustering, a network feature analysis of a manageable but representative subgraph is initially performed. The main objective of this analysis is to classify each of the edges included as either "community" or "non-community". Obviously, this edge labelling procedure transforms this subgraph to a supervised machine learning-ready dataset that will be given as training input to a distributed logistic regression model. This trained community prediction model will be finally applied in terms of the original graph's community structure identification. The experimentation with various real-world social networks verifies that the proposed method, is remarkably promising, highly scalable and noticeably efficient for community detection despite slightly sacrificing part of original Louvain's accuracy.
The remainder of this manuscript is organized as follows:

- In Section 2 the related work on community detection methodologies is given.
- In Section 3 the proposed methodology and its implementation is comprehensively presented.
- In Section 4 the experimentation results are discussed.
- In Section 5 the conclusions and the future work are provided.

2 Related Work

The community detection significance has attracted the interest from various scientific fields. Copious methods from different backgrounds and objectives have been proposed to detect the underlying community structure of a given graph by leveraging its topological information. There is a plethora of algorithms and methodologies [1] [2] [20], which can be roughly distinguished as divisive algorithms, agglomerative algorithms, transformation methods and other approaches accordingly.

2.1 Divisive Algorithms

The fundamental idea of divisive algorithms is to identify and remove all the edges that interconnect nodes belonging to different communities by applying an iterative process. At the beginning, the original graph is considered a single community. During each iteration step, a set of edges that meet certain criteria and appear to interconnect different communities, will be removed. The process is repeated until the retrieved network structure converges to a stable state where no additional edge removal can be applied [20]. It is worth mentioning that in a few extreme cases the removal of a whole subgraph may be required in terms of community structure validity and efficiency.
The most representative divisive algorithm is the one proposed by Girvan and Newman [18], in which the removal criterion depends on the edge betweenness measure [9].

Despite, its algorithmic and conceptual simplicity, this approach is excessively demanding in both calculation and storage resources requirements and hence considered impossible for big data networks.
Ample alternatives of Girvan and Newman algorithm have been proposed aiming to substitute the original edge betweenness modularity, such as geodesic or current-flow edge betweenness etc. [1] [2], in terms of efficiency increase or complexity reduction. Among the proposed, the most prevalent is the random-walk edge betweenness criterion [3] [6], which identifies low density edges that are not part of cycle paths and can be considered as community connection edges.

2.2 Agglomerative Algorithms

In contrast to divisive algorithms, the agglomerative are classified as bottom up approaches. By originally considering each node as a separate community, also known as singleton, a number of iterations is applied, in each of which, the distinct communities are merged according to the calculated result of a well-defined similarity function. The ultimate goal is to end up with a graph that compose a unique cluster [1].
It is worth pointing out that agglomerative algorithms intrinsically unfold the complete hierarchical community structure of the analysis graph, as new communities are built during the process from previous steps' communities. Even if the generated hierarchies might seem generally artificial, social networks are typical examples of hierarchical structured information since it is natural for people to create groups within their working environment, friends or even families.
Among the proposed agglomerative algorithms [2] [20], the one introduced by the university of Louvain [4] is the most widespread and considered as the state-of-the-art in terms of hierarchical community detection. The significant difference of this greedy approach is that the modularity measure used, overtake the rests in terms of simplicity and accuracy. In Louvain's algorithm [4], the modularity function ingeniously combines the number of community's intra-connecting edges, the corresponding nodes' degrees and the total number of graph's edges, aiming to identify a final structure with no edges between the densely inter-connected retrieved communities.
Extensive research has been done regarding Louvain's alternatives, where the most notable are either trying to combine Louvain's modularity function with other modularity alternatives, as the one presented in [5] that uses Infomap, or to parallelize the structure retrieval procedure [10] [23].

2.3 Transformation Methods

Many researchers of different backgrounds have been engaged to tackle this critical graph clustering problem by trying to transform and project the initial network structure identification problem to a different solution space. The most characteristics are the Spectral Clustering techniques and the adoption of Genetic Algorithms.
The Spectral clustering [1] techniques are following the transformation methodology where each node is represented by a point in space, whose coordinates are elements of eigenvectors. Consequently, simple clustering techniques, such as K-means clustering,

can be applied to come up with the clusters that actually are the original graph's communities [14] [15]. The main drawback of this approach is that the clustering results are seriously poor for graphs originally comprised of many fragmented communities.
Another widely used set of algorithms, in terms of modularity optimization, is the Genetic algorithms [7]. Those are particularly repetitive meta-heuristics inspired by the theory of natural evolution. As in the natural evolution process, the original graph is randomly initialized, and each node is assigned to a community [20].

2.4 Other Approaches

Any of the previously described algorithms try to reveal the underlying community structure for any given network. Although, by only depending on graph's topological structure without taking into account the critical user content information, it is unquestionable that a fundamental piece of social network information is being ignored. As mentioned in [1] and [20], many approaches have been proposed, such as [12], that try to take leverage the user content information to proceed to a more beneficial graph clustering outcome for user-oriented applications.
Additionally, another engaging alternative is the link prediction methodology. This approach's initial intention was the prediction of potential connections between any pair of users that are not currently connected, based on the existing user relationships. However, recent studies [9] [11] showed that link prediction can also be efficiently used for extensive graph analysis due to its limited computation and resources requirements. Therefore, by slightly modifying the original scope of link prediction, it is obvious than this methodology can equivalently be used to predict whether any pair of directly connected nodes, belong to the same community or not, which is the basic idea of this paper.
Other equally important algorithms and methodologies such as Simulated Annealing [1], Information Diffusion Graph Clustering [20], External Optimization [1] and Generative Models [2] are not further discussed since they are out of the scope of this paper.

3 Proposed Methodology

3.1 Graph Statistical Analysis & Feature Enrichment

This is the initial processing step where complementary to graph's statistical analysis, a predefined set of features is assigned to each graph's node and edge accordingly.
The statistical analysis regarding the original graph's topology structure is mandatory in order to ensure that the consequently extracted subgraph would be representative to the original. As a result, the average node degree, which is the average number of connections per node, and the graph's degree distribution, which is the probability distribution of node degrees over the whole network, need to be calculated.
Regarding the feature enrichment process, a predefined set of features, separately per node and edge, is calculated and assigned accordingly. It is worth pointing out that the amount of information implied only by the edge and its adjacent nodes, might not be sufficient for proper prediction. Undeniably, each connection's properties are not only

affected by their direct adjacent nodes, but also substantially influenced by their surrounding network structure. Hence, the current level predictors should be enriched with the values of all of the corresponding features up to a predefined depth k.
Empirically, the features that correspond to depths greater than 5, do not add any essential information since they tend to converge to graph's average values. On the contrary, the experimentation process, which is comprehensively analyzed in the following subsection, has practically proven that the community prediction is substantially benefited with feature enrichment up to the 3rd connection level.
The only node-oriented feature calculated is the number of k-depth neighbors. Since this algorithm's ultimate scope is the edge-oriented prediction, all calculated feature's values for each engaging node will be included as impeding edge's additional information.
The edge-oriented prediction features are:

- The loose similarity: regarding an imminent pair of nodes and each of their corresponding sets of distinct nodes up to depth k, this is the fraction of common over the union set's total number of nodes. It should be underlined, that both sets include nodes having depth less or equal to k, since a common node might be at a different depth for each impeding node.
- The dissimilarity: regarding an imminent pair of nodes and each of their corresponding sets of distinct nodes up to depth k, this is the fraction of uncommon over the union set's total number of nodes.
- The edge balance: regarding an imminent pair of nodes and each of their corresponding sets of distinct nodes up to depth k, this is the fraction of the absolute difference between the peers adjacent over the union set's total number of nodes. This feature values ranges from 0 to 1, with values closer to 0 indicating a balanced edge that have merely equal k-depth vertices on both sides.
- The edge information: considering that the corresponding element on a graph's kth power adjacency matrix indicates the number of distinct k-length paths between the impeding vertices [21], this feature reveals the pair's interconnection strength.

3.2 Representative Subgraph Extraction

To properly train a community prediction model, the knowledge of Louvain algorithm's behavior regarding the original graph is considered radical. Nevertheless, considering the size of real-world social graphs, it is obvious that even the execution of Louvain's distributed implementation [23] can be considered prohibitive, since it should still need to repetitively check an immense number of candidates per iteration in order to merge the ones with the highest probability of forming a community. Thus, the extraction of a subgraph that retain the original graph's elementary topology properties is deemed fundamental and that is indeed the main purpose of this processing step.
After thorough investigation, the Tiny Sample Extractor algorithm [17] has been selected for its well efficiency and high scalability. Even if the extraction of a representa-

tive subgraph yet remains controversial, since the real networks exhibit the degree distribution that follow the power law, the extracted subgraph's degree distribution remains remarkably similar to the original graph's, as clearly shown in Figure 1.

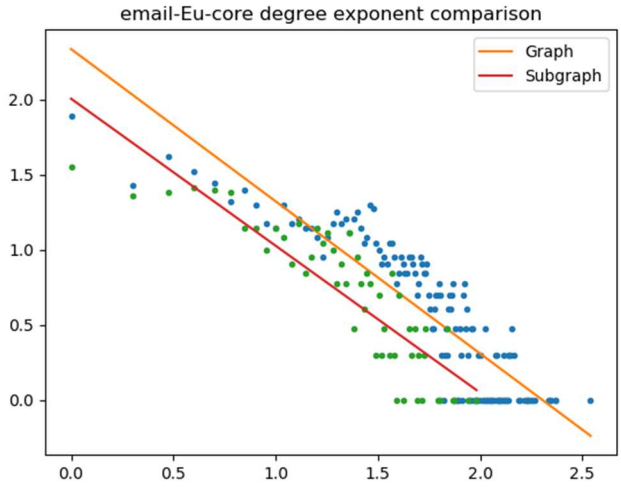

Fig. 1. The exponent degree distribution on log-log scale for Email-Eu-Core [27] dataset.

Having already calculated the original graph's degree distribution during the previous graph analysis processing step, the Tiny Sample Extractor [17] algorithm initially tries to estimate the original graph's exponent degree by the slope of the log-log Complementary Cumulative Distribution Function (CCDF). Bear in mind, that the CCDF function for a given degree value d, returns the fraction of nodes that have degree greater than d. Therefore, starting from a random node, 3 biased random walks are applied in turn. In the first couple of walks, the minimum and maximum exponent degree values are respectively defined. In the last one, the resulting subgraph is extracted. The bias included is used in terms of compensation due to the skewed distribution of nodes visited in the random walks performed.
In the extreme case, where the original graph is that big, in such extent that is not even traversable, then both the previously described statistical analysis and the representative subgraph extraction step can be applied only to a smaller part of the original graph that would is considered as the graph's "known" part.

3.3 Community Prediction Model Training & Application

The most suitable machine learning model for prediction, in terms of big data efficiency and interpretability, is undoubtedly the logistic regression model. Projecting its application to community detection, the eventual scope of this model would be to predict whether a given edge connects nodes of the same community or not, depending on the aforementioned edge-oriented graph topology features. Despite the intrinsic accuracy

drawbacks of linear models, it is frequently proven that on real-worlds problems, they seem to be surprisingly competitive to non-linear thanks to their low variance.
To appropriately train a logistic regression model, the classification response needs to be predefined at the training dataset. Among the copious proposed methods, the Louvain's one [4] is broadly considered [6] [20] as the state-of-the-art in disjoint community detection, while its performance generally serves as the golden baseline. So, using the previously extracted subgraph, the distributed version of Louvain algorithm [23] is applied to classify all subgraph's edges as "community" or "non-community".
As it is broadly known, since Louvain's returned community structure is excessively affected not only by the traversal fashion (BFS, DFS) but also from the starting candidate that the execution starts, it would be prudent not only to re-execute it starting from different nodes but also to use different traversing methodologies. Even if this repetition might sound extremely demanding, the size of the extracted subgraph significantly restricts the required calculations.
The large number of descriptive topology features supposed to improve the model prediction accuracy. However, it is very usual for some or many input features to either not be truly associated with the response or to be highly correlated with each other. The first case, also known as the "curse of dimensionality", refers to the situation where irrelevant features are included during training, while the second, also known as the "collinearity effect" occurs when two or more predictors are correlated to one another. Both effects lead to unnecessary complexity introduction and interpretability reduction. Consequently, during the training process the model should automatically proceed to feature selection by applying the L1 regularization penalty that excludes the redundant features and practically performs feature selection.
Consequently, using the Spark MLlib's [22] distributed libraries, a L1 regularized distributed logistic regression classifier is trained, using the previously extracted subgraph as training dataset and the enrichment features described on the first processing step as predictors. This model prediction is used to identify the original graph's underlying community structure.

4 Experiments

To evaluate the performance of the proposed methodology, its predicted structure is compared against the communities returned from the distributed Louvain implementation [23]. For this experimentation procedure, the following widely used and thoroughly analyzed social graphs are evaluated. It is noteworthy that the first two [24] [25] datasets considered reference datasets, since they have historically attracted the scientific interest and have exhaustively been analyzed.

Table 1. Evaluated Datasets

Dataset	# of Nodes	# of Edges	Edges / Nodes
Zachary karate club [24]	34	78	2.29
Dolphins [25]	62	159	2.56

Hamster friendships [26]	1858	12534	6.75
email-Eu-core [27]	1005	25571	25.44
Bitcoin Alpha [28]	3738	24186	6.39
Enron email [29]	36692	183831	5.01

All the experiments were executed in an eight node Spark 2.2.0 cluster, with 4 GBs RAM and 1 virtual core per each virtual machine. Even if the processing performance is out of the scope of this comparison, it is worth mentioning that the proposed algorithm's processing time and memory requirements were constantly less, at least 25% and 55% respectively, comparing to the distributed Louvain implementation [23]. The performance improvement will certainly be substantially higher in bigger social graphs, since the proposed methodology's complexity is linear and not of a higher polynomial degree as in Louvain's case.

In terms of completeness, all the elementary classification performance metrics have been calculated. Specifically:

- The accuracy, which is the number of edges correctly classified as either "community" or "non-community" over all the predictions made.
- The precision, which is the number of edges correctly classified as "non-community" over the total number of "non-community" predictions.
- The recall/sensitivity, which is the fraction of edges correctly classified as "non-community" over the total number of truly "non-community" edges.
- The specificity, which is the fraction of edges correctly classified as "community" over the total number of truly "community" edges.

The community detection results regarding the aforementioned social graphs are presented in the following table, Table 2.

Table 2. Classification Performance Metrics

Dataset	Accuracy	Precision	Recall	Specificity
Zachary karate club [24]	76.5%	100%	50%	100%
Dolphins [25]	88.2%	91.7%	78.6%	95%
Hamster friendships [26]	69.7%	72.3%	54.5%	82.5%
email-Eu-core [27]	78.1%	81.8%	80.2%	75.2%
Bitcoin Alpha [28]	89%	92.6%	14%	99.8%
Enron email [29]	82%	73.2%	27.7%	97.2%

As it is shown in Table 2 the proposed prediction model shows excellent overall prediction performance with splendid statistics on precision and specificity.

Specifically, the trained prediction model shows magnificently rational general accuracy that reach the 80.6% in the average case. In other words, the proposed algorithm remarkably identifies each edge's class in general.

Focusing on the outstanding specificity and precision statistics, it is more than obvious that the proposed implementation impressively identifies the "community" edges. This

way, the graph's consistency is preserved by retaining social graph's true connections and avoiding graph's fragmentation.
However, the mediocre recall statistics shows that the trained model misclassifies several "non-community" edges as "community" ones. This will doubtlessly lead to a less fragmented community structure comparing to the one returned from original Louvain's algorithm. Actually, this can be graphically confirmed by the following figure, Figure 2 where the Email-Eu-Core [17] graph's returned community structure is presented in both cases, having the red colored edges stand for the "non-community" edges and the blue ones for the "community".

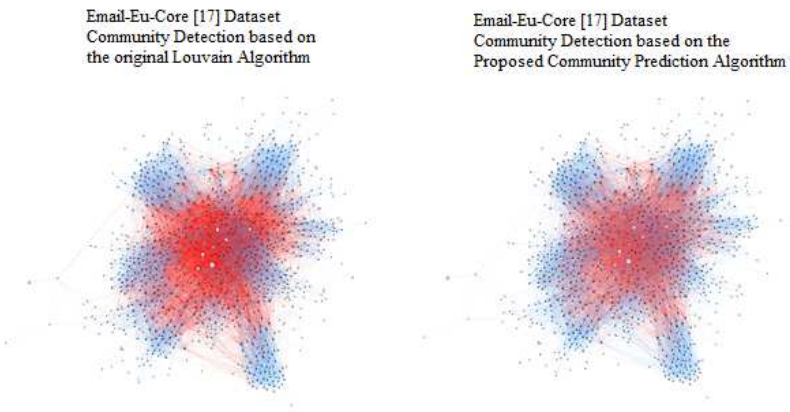

Fig. 2. Email-Eu-Core [17] Dataset Community Detection comparison

As presented, the returned community structure is eminently similar in both cases. The "non-community" edges are apparently less in the proposed community prediction algorithm situation. However, the use of more complex machine learning models, such as non-linear prediction ones, is supposed to catapult the recall statistics, since linear prediction models inherently focus on interpretability and not on prediction's accuracy.

5 Conclusions

In this manuscript, a novel distributed community prediction methodology has been introduced based on Louvain's [4] community detection algorithm. Unquestionably, the experimentation process confirmed the remarkable performance in identifying the subjacent community structure of social graphs comparing to Louvain's original algorithm.
Despite the encouraging results, it is obvious that the methodology's community prediction can be substantially improved by:

- Applying a graph cleansing processing step, where the outliers' removal will be applied.

- Combing the collinear variables into a single predictor to tackle the multi-collinearity effect.
- Applying less interpretable but more accurate models, e.g. non-linear prediction models.

But all the above are left for future work.

Acknowledgment

Christos Makris is co-financed by the European Union (European Social Fund) and Greek national funds through the Operational Program “Research and Innovation Strategies for Smart Specialization - RIS3” of "Partnership Agreement (PA) 2014-2020".

References

1. Santo Fortunato: Community detection in graphs, Physics Reports 486, 75-174 (2010).
2. Satu Elisa Schaeffer, Graph clustering, Computer Science Review I, 27-64 (2007)
3. Liu, X., et al.: MIRACLE: A multiple independent random walks community parallel detection algorithm for big graphs. Journal of Network and Computer Applications (2016), http://dx.doi.org/10.1016/j.jnca.2016.05.008i
4. Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre: Fast unfolding of community hierarchies in large networks, Journal of Statistical Mechanics Theory and Experiment, CoRR abs/0803.0476 (2008).
5. Held P., Krause B., Kruse R.: Dynamic Clustering in Social Networks using Louvain and Infomap Method, Third European Network Intelligence Conference (2016)
6. Partha Basuchowdhuri, Varsha Nagarajan, Khusbu Mishra, Satyaki Sikdar, Surabhi Gupta, Subhashis Majumder: Fast Detection of Community Structures using Graph Traversal in Social Networks. CoRR abs/1707.04459 (2017)
7. Jianhai Su, Timothy C. Havens: Fuzzy community detection in social networks using a genetic algortihm. FUZZ-IEEE 2014: 2039-2046 (2014)
8. Amato G., Candela L., Castelli D., et al.: How Data Mining and Machine Learning Evolved from Relational Data Base to Data Science, A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years, Studies in Big Data 31, Springer International Publishing AG (2018), DOI 10.1007/978-3-319-61893-7_17
9. William Cukierski, Benjamin Hamner, Bo Yang: Graph-based features for supervised link prediction. IJCNN 2011: 1237-1244 (2011)
10. Mahmood Fazlali, Ehsan Moradi, Hadi Tabatabaee Malazi: Adaptive parallel Louvain community detection on a multicore platform. Microprocessors and Microsystems - Embedded Hardware Design 54: 26-34 (2017)
11. Jun Pang, Yu Gu, Jia Xu, Ge Yu: Semi-supervised multi-graph classification using optimal feature selection and extreme learning machine. Neurocomputing 277: 89-100 (2018)
12. Ru Wang, Seungmin Rho, Wandong Cai: High-performance social networking: microblog community detection based on efficient interactive characteristic clustering. Cluster Computing 20(2): 1209-1221 (2017)
13. Liang Bai, Xueqi Cheng, Jiye Liang, Yike Guo: Fast graph clustering with a new description model for community detection. Inf. Sci. 388: 37-47 (2017)

14. Xiaolong Deng, Jiayu Zhai, Tiejun Lv, Luanyu Yin: Efficient Vector Influence Clustering Coefficient Based Directed Community Detection Method. IEEE Access 5: 17106-17116 (2017)
15. Cem Aksoylar, Jing Qian, Venkatesh Saligrama: Clustering and Community Detection With Imbalanced Clusters. IEEE Trans. Signal and Information Processing over Networks 3(1): 61-76 (2017)
16. Andreas Kanavos, Isidoros Perikos, Ioannis Hatzilygeroudis, Athanasios K. Tsakalidis: Emotional community detection in social networks. Computers & Electrical Engineering 65: 449-460 (2018)
17. Harish Sethu, Xiaoyu Chu: A new algorithm for extracting a small representative subgraph from a very large graph. CoRR abs/1207.4825 (2012)
18. Mark E.J. Newman, Michelle Girvan: Finding and Evaluating Community Structure in Networks. Physical review. E, Statistical, nonlinear, and soft matter physics. 69. 026113. 10.1103/PhysRevE.69.026113. (2004)
19. Wangsheng Zhang, Gang Pan, Zhaohui Wu, Shijian Li: Online Community Detection for Large Complex Networks. IJCAI 2013: 1903-1909 (2013)
20. Bisma S. Khan, Muaz A. Niazi: Network Community Detection: A Review and Visual Survey. CoRR abs/1708.00977 (2017)
21. Kranda, Dan: The Square of Adjacency Matrices., arXiv:1207.3122, (2012)
22. Xiangrui Meng, Joseph K. Bradley, Burak Yavuz, Evan R. Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, D. B. Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, Ameet Talwalkar: MLlib: Machine Learning in Apache Spark. Journal of Machine Learning Research 17: 34:1-34:7 (2016)
23. Sotera Distributed Graph Analytics (DGA): Sotera Defence Solution: https://github.com/Sotera/spark-distributed-louvain-modularity.git
24. Zachary karate club network dataset -- KONECT, April 2017.: http://konect.uni-koblenz.de/networks/ucidata-zachary
25. Dolphins network dataset -- KONECT, April 2017.: http://konect.uni-koblenz.de/networks/dolphins
26. Hamster friendships network dataset -- KONECT, April 2017.: http://konect.uni-koblenz.de/networks/petster-friendships-hamster
27. B. Klimmt, Y. Yang. Introducing the Enron corpus. CEAS conference, 2004.: https://snap.stanford.edu/data/email-Enron.html
28. Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. "Local Higher-order Graph Clustering." In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017.: https://snap.stanford.edu/data/email-Eu-core.html
29. S. Kumar, F. Spezzano, V.S. Subrahmanian, C. Faloutsos. Edge Weight Prediction in Weighted Signed Networks. IEEE International Conference on Data Mining (ICDM), 2016.: https://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html