A Practical Comparison of Local Graph Clustering Algorithms

Rashed Khalil Salem[1],Wafaa Tawfik Abdel Moneim[2],Mohamed Monir Hassan[2]
[1]*Faculty of Computers and Information, Menoufia University, Egypt*
[2] *Faculty of Computers and Informatics, Zagazig University, Egypt*

Abstract
Nowadays a large number of applications of graph clustering are available, with expanding the span of the graph the conventional methods of clustering is not appropriate to manipulate these graph because it is costly for computation. Local graph clustering algorithms solve this problem by working on a given vertex as input seed set without looking at the whole graph to find a good cluster. The conventional algorithms are slower than the local clustering algorithms. In this paper, we show a comparison between two of local graph clustering algorithms are HK-relax and SimpleLocal based on conductance and runtime. We display experiments on large-scale graphs and showing that SimpleLocal finds a good cluster with a small conductance that HK-relax but this take more runtime. We also show the seed set size effect on two algorithms as input parameter and find that large size of the seed set gives a good conductance than a small seed set size. In addition to display locality parameter influence on SimpleLocal as input, from the outcomes, we recognize that with decreasing the value of locality δ there is a good conductance of graph clustering.

Keywords: *Data Mining, Graph Mining, Dig Data, Graph Clustering, Local Graph Clustering*.

I. INTRODUCTION

Datasets are identified by the big data term according to large size and complexity. The traditional techniques such as data mining techniques cannot handle big data. Extracting useful knowledge or hidden pattern from these large dataset based on its volume, variety, velocity, value, and veracity is called big data analytics this is a challenge of big data [1]. Graph databases are increased in everywhere. Structure relationships between objects build a graph model. There are many applications for graph model such as social networking, biology, chemistry, image processing, web link analysis, computer networks, and human genome assembly. Graph mining is the process of dealing with graph data by using data mining and machine learning techniques to detect useful and unexpected patterns [2]. Big graph mining is the process of extracting meaningful information from large amounts of graph data which reaches Tera and Petabyte[3].

Clustering is the unsupervised procedure. The process of splitting a set of input data into two categories called clustering. Based on similarity measurements, similar contains the objects within the same clusters and dissimilar contains objects from various clusters. Graph clustering is the process of dividing the vertices in a graph into groups based on there is inside the group high edge density and outside the group low edge density. A cluster is a group of vertices. Partition vertices into connected subgraphs also called graph clustering which there are more connections between the vertices in the same cluster and fewer connections with a various cluster.

Review papers [4], [5] introduce various graph clustering algorithms. Using the whole graph as input for the clustering process called global graph clustering and using a certain seed vertex for the clustering process called local graph clustering.There are many applications of graph clustering such as correlation clustering, graph partitioning, community detection, a protein-protein network, etc.Many algorithms of global clustering are described in [6,7,8, 9, 10, 11,12].

Graph clustering traditional algorithms need the whole graph for processing that is very expensive computations so that this paper focuses on local graph clustering algorithms. Local graph clustering algorithms based on vertices number and/or edges number in the input seed set or output cluster during the running time. Conductance is used to measure the connectivity between vertices in a graph. Set conductance is calculated by the ratio of edges number leaving the set to the number of the edges touched by the set of vertices.The small conductance value stands for many internal edges within the set and few edges outside it.An efficient cluster is chosen by a subset of vertices whose inner connections are larger than its outer connections.

The paper contains a comparative study of local graph clustering algorithms. These algorithms are heat kernel [13] and SimpleLocal [14] to identify small conductance in a network.We compare them based on conductance and time as well as the seed size effect on conductance. In section 3 we begin by providing related work. Section 4 represents the preliminaries of the graph. Section 5 discusses the local graph clustering algorithms. We describe the experiments in section 6. The conclusions are shown in section 7.

I.

II. RELATED WORK

Study local algorithms for large graphs have numerous papers,Spielman and Teng [15, 17] introduce an algorithm called Nibble for solving symmetric linear systems using nearly-linear time algorithms for graph partitioning. Andersen, R., et al. [16] develop an algorithm for computing approximate PageRank vectors to find cuts with nearly optimal conductance which runs in time $O(2^b \log^4 m/\phi^5)$, this algorithm called PageRank-Nibble.

Finally, Fountoulakis, K., et al. [18]present new trends of optimization on local graph clustering of the PR-Nibble algorithm.Andersen and Lang [19] introduce an algorithm for improving the graph partitioning called Improve with a subset of vertices as input to produce the best set.

Andersen and Peres [20]presenta random algorithm to find a sparse cut by emulating the volume-biased evolving set process. Kwok and Lau [21] also,solve a small sparsest cut problem by utilizing bicriteria approximation algorithms. Many graph clustering algorithms for community discovery problem include Metis [22], Graclus [23] and Markov clustering [24].

There is a modified version of PageRank that based on two arguments are a seed and a temperature or heat constant called heat kernel PageRank that is designed by Chung [25]. An exponential sum of random walks from the seed can be used to express the heat kernel PageRank, scaled by the temperature.

The diffusion is computed by the first deterministic local algorithm called HeatKernel-relax for studying the communities introduced in [13]. This algorithm is a relaxation method that evaluates the matrix exponential to solve the linear system as well as comparing this algorithm with PageRank. The heat kernel executes superior to the PageRank diffusion for communities.

Orecchia and Zhu [26] get the best approximation guarantee by combining spectral and flow methods for local graph clustering, this is the first strongly local flow-based method, that supply two local algorithms, LocalFlow and LocalFlowexact.

Veldt, et al. [14] introduce SimpleLocal algorithm for locally-biased graph-based learning. The advantages of this algorithm are strongly-local this means it is not based on the whole graph to extract good conductance cuts. This algorithm uses an implicit ℓ_1-norm penalty term to achieve the localization. SimpleLocal runtime is weaker than Orecchia and Zhu [26].

Yin and Benson [27] present method of local graph clustering depend on the higher-order network (network motif). This method called Motif-based Approximate Personalized PageRank (MAPPR) algorithm that detects clusters with lower motif conductance. This method is fast and effective for directed graphs.

III. LOCAL FLOW-BASED METHOD

The SimpleLocal [14] uses the existing max-flow algorithms to introduce a new strongly-local flow algorithm. This algorithm is flexible and easy to implement and solves the same optimization problem as LocalImprove.

A three-stage method is improved for exact maximum flow computations on $G'_R(\alpha, \delta)$ instead of using Dinic's algorithm to compute approximate maximum flows.

A. Three-Stage Local Max Flow Procedure

3StageFlow is used to compute a maximum s-t flow of a modified augmented graph $G'_R(\alpha, \delta)$. Local graph L = (N_L, E_L) is a subset of the modified augmented graph $G'_R(\alpha, \delta)$ that includes

- Add *s*, *t* to the graph G' .
- Edges from *s* to the *R* set.
- Edges between nodes in *R*.
- Edges from *R* to Neighbor(*R*).
- Edges from *t* to the Neighbor(*R*).

F is a flow vector that starts with zero vector, and flow (*F*) is an aggregated amount of all flow from *s* to *t*. Figure 3 describes the 3StageFlow flowchart.

Step 1. Expansion

For much flow in the local graph from *s* to *t*, there is needed to expand graph at the start of each iteration. The expanded set of vertices is indicated by *X*.

Step 2. Max-Flow Computation

After the first step is completed correctly, maximum flow *f* is calculated by using max-flow subroutine. The value of *F* is updated to *F* + *f*. The structure flow residual graph L_f, is computed.

Step 3. Updates

This step is used to analyze the flow effects and decide if there is a need to expand the local graph. The residual graph of *f* is used to develop the local graph to discover an unsaturated edges chain of the vertices set that remain connected to *s*, these vertices set called the source set *S*.

Algorithm 1: 3StageFlow
Input: graph *G*, parameters α, δ, seed set *R*
Initialize:
V_L := {R, Neigh(R), s, t}
E_L := {(s, R), (R, Neigh(R)), (Neigh(R), t)}
F := 0; X := ϕ
while X ≠ ϕ; or F = 0 **do**
1. Expand W
for x ∈ X do
$V_L \leftarrow V_L \cup$ Neigh(x)
$E_L \leftarrow E_L \cup \{(x, v) : v \in V_L\} \cup \{(y, t) : y \in$ Neigh(x)$\}$
end for
2. Max Flow:
$f \leftarrow$ MaxSTflow(L); $F \leftarrow F + f$
3. Update L
$L \leftarrow L_f$; $S \leftarrow$ source set
$X \leftarrow$ vertices whose edge to *t* was saturated
end while

Fig. 3 3StageFlow flowchart.

B. SimpleLocal Algorithm

A good conductance cut of SimpleLocal is computed by calling 3StageFlow repeatedly to detect the smallest α like that the maximum *s-t* flow of G'_R (α, δ) is less than αvol(*R*). Figure 4 illustrates the SimpleLocal flowchart.

Algorithm 2 SimpleLocal
Input: *G*, *R*, locality parameter δ ≥ 0
α := ϕ(R)
[F; S] := 3StageFlow G'_R (α, δ)
while flow(F) < α vol(R) **do**
α ← ϕ(S); S^* ← S
[F, S] := 3StageFlow G'_R (α, δ)
end while
Return: S^*

Fig. 4 SimpleLocal flowchart.

IV. HEAT KERNEL ALGORITHM

The heat kernel [13] is a deterministic approach, begin with seed vertices for recognizing a community, this is a type of graph diffusion. An adjacency matrix is represented by A and D is the diagonal matrix of degrees that is calculated by D_{ii} = di. The random walk transition matrix is computed by $W= (D^{-1}A)^T= AD^{-1}$.A graph diffusion is calculated by the equation

$$df = \sum_{i=0}^{\infty} \alpha_i W^i s \quad (1)$$

where $\sum_i \alpha_i$=1 and s is a stochastic vector. A small conductance community is calculated by a sweep procedure using a diffusion f estimatefrom a seed.

The heat kernel equation substitutes $\alpha_k$with$t^i/ i!$

$$hk = e^{-t}\left(\sum_{i=0}^{\infty} \frac{t^i}{i!}(W)^i\right) s = \exp\{-t(I-W)\}s \quad (2)$$

This algorithm called HK-relax because of using coordinate-relaxation method for approximating h to execute this, first approximate exp {*tW*} with its degree *N* Taylor polynomial, *TN*(*tW*) then compute *TN*(*tW*)*s*. An equation that uses Taylor polynomial to compute an approximation for a matrix *G* is:

$$\exp\{G\} = \sum_{i=0}^{\infty} \frac{1}{i!} G^i \approx \sum_{i=0}^{N} \frac{1}{i!} G^i \quad (3)$$

The Pseudo-code for the HK-relax algorithm is presented in figure 5.

V. EXPERIMENTAL RESULTS ANDANALYSIS

In this part, the experimental results for HK-relax and SimpleLocal on a group of graphs. Two algorithms compute the runtime and conductance beginning with the seed set. The required parameters for HK-relax are *t* and ε, and the locality parameter δ for SimpleLocal. As well as explaining the locality parameter and seed set size effects of SimpleLocal on social networks. In addition to studying the seed set size effects of HK-relax on communities. The experiments are performed on CPUsystem with Intel(R) Core(TM) i7-4510U (2.60 GHz), 8.00GB of memory, and Windows 10 64-bit Operating System.

```
Input: graph G
parameters: seed, t, eps, N, psis
Initialize: residual r, queue Q
for s ∈ seed
    r[(s ,0)] ← 1./ len (seed )
    Q. append ((s ,0))
End For
while len(Q) > 0 do
  (v, j) ← Q. popleft
  rvj ← r[(v, j)]
  If v ∉ x Then
     x[v] += rvj
     r[(v, j)] = 0
     Mass = (t*rvj / (float (j) +1.))/ len (G[v])
  End If
  For u ∈ G[v]
     next ← (u,j+1)
     If j+1 == N Then
        x[u] += rvj /len(G(v))
        continue
     If next ∉ r Then
        Thresh ← math .exp(t)* eps*len(G[u])
        Thresh ← thresh / (N*psis [j +1])/2.
     End If
     If r[ next ] < thresh and r[ next ] + mass >= thresh Then
        Q. append( next )
     End If
     r[ next ] ← r[ next ] + mass
  End For
End While
```

Fig. 5 Pseudo-code of HK-relax algorithm.

A. Datasets

This paper performed experiments on 11 datasets [28]. These datasets are undirected graph collected from the communities and a collaboration network and described in table1.

TABLE 1: GRAPH DATASETS.

Graph	|V|	|E|
Erdos02	6927	16944
Hep-th	8361	31502
Ca-HepTh	9877	51971
Ca-HepPh	12008	237010
Ca-AstroPh	18772	396160
As-22july06	22963	96872
Cond-mat-2003	31163	240058
Cond-mat-2005	40421	351382
Usroads-48	126146	323900
Com-DBLP	317080	2099732
Com-Amazon	334863	1851744

B. Runtime and conductance

The heat kernel rank is computed for four different parameter sets (t, ε) = (10, 10−4); (20, 10−3); (40, 5 *10−3); (80, 10−2) and produce the best conductance between them and the SimpleLocal for locality parameter δ = 0.1. Figure 6 describes the conductance results of SimpleLocal and HeatKernel. Figure 7 shows the runtime results of SimpleLocal and HeatKernel.

These results show that SimpleLocal produces a small conductance value than the HK-relax algorithm that produces. A small conductance means a good cluster. This suggests that SimpleLocal algorithm better than the HK-relax algorithm for local graph clustering. The result of the runtime of two algorithms shows that SimpleLocal takes more time than HeatKernal algorithm when computing conductance value.

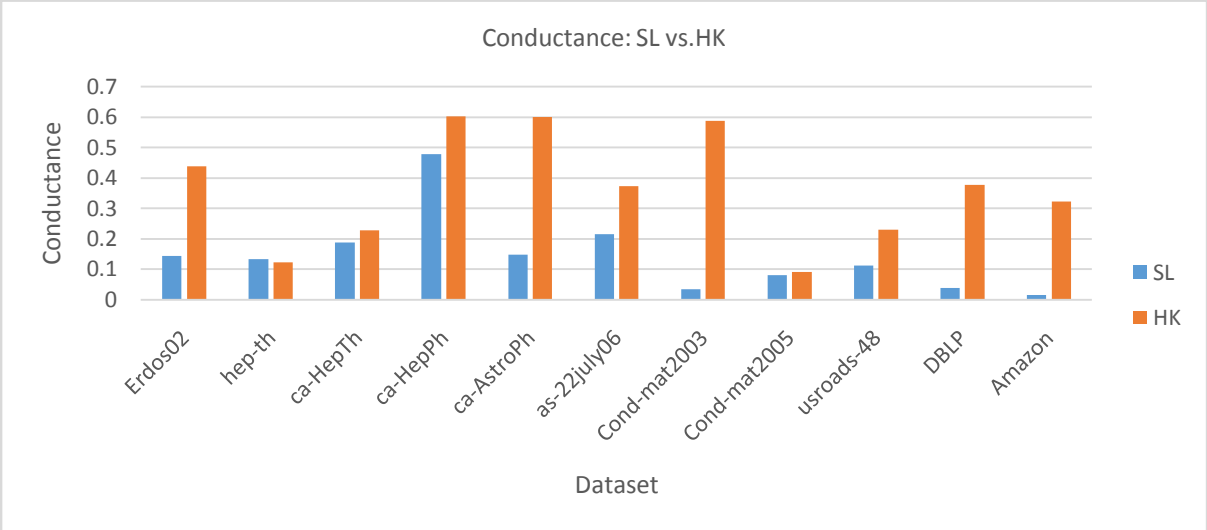

Fig. 6 The conductance of SimpleLocal and HeatKernel.

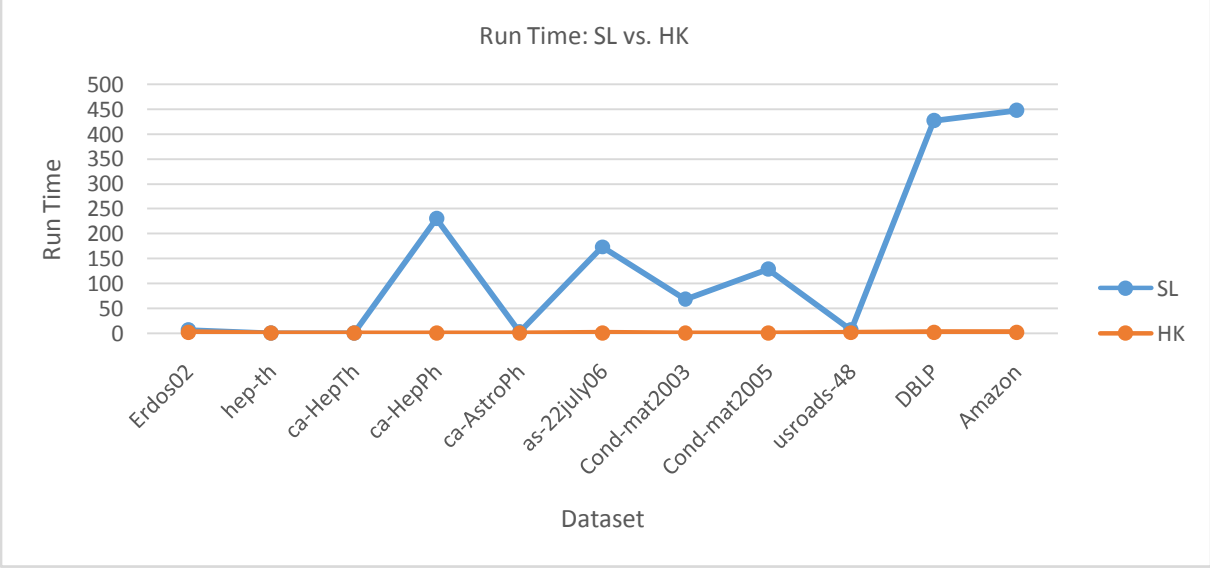

Fig. 7 *Runtime of SimpleLocal and HeatKernel*

Locality parameter effect

Study the locality parameter effect on the SimpleLocal algorithm for finding a good cluster of the graph. The experiments performed on a cond-mat-2003 dataset for increasing values of δ from 0 to 1. Figure 8 shows the locality parameter effect on the SimpleLocal algorithm.

After doing the experiments, increasing the locality parameter value lead to decrease the conductance value of SimpleLocal that stands for getting a good conductance cut with high efficiency of the cluster.

Seed size effect

This section shows the seed size effect on the conductance of the communities. The seed set is the input of the HeatKernel algorithm. The experiments do in a com-DBLP dataset. Figure 9 describes the influence of seed set size on the conductance of

HeatKernel algorithm. From the results, it is obvious with increasing the seed set size. This means that to obtain the good conductance cut, there is a need to increase the size of the seed set.

As well as, show the size of the seed set on the conductance of SimpleLocal. The experiments are performed on Amazon dataset. Figure 10 describes that the conductance value of HeatKernel reduced the effect of the size of the seed set on the conductance of SimpleLocal algorithm. After displaying the results, the seed size set is inversely proportional to the conductance of SimpleLocal, this means that, to get a good cluster use large seed set as input.

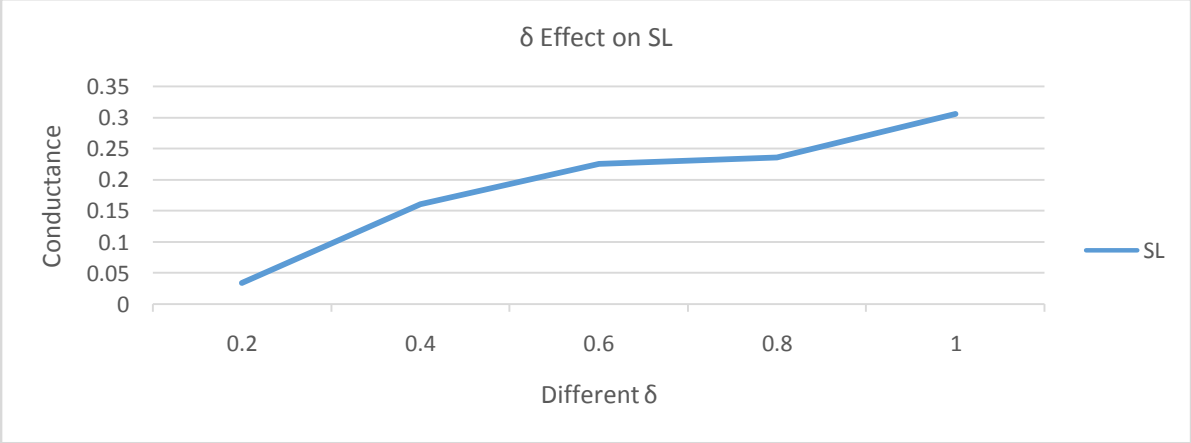

Fig. 8 The locality parameter effect on the conductance of SimpleLocal.

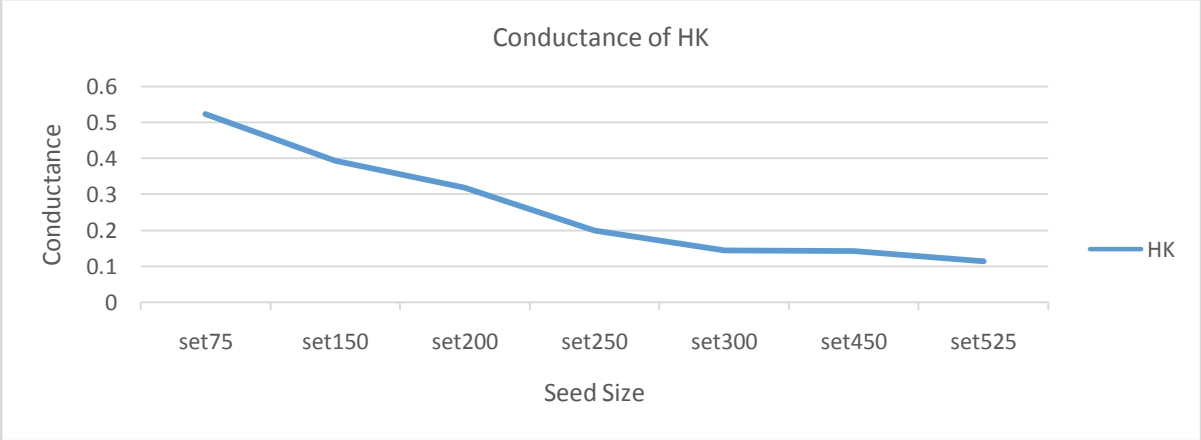

Fig. 9 The effect of a seed set size on the conductance of HeatKernel.

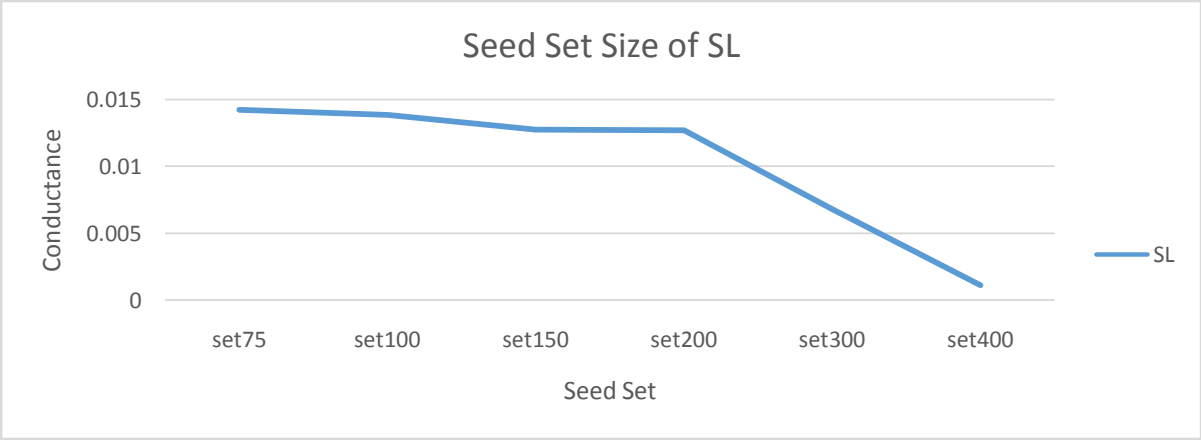

Figure 10: The effect of a seed set size on the conductance of SimpleLocal.

VI. CONCLUSION

These results suggest that the SimpleLocal algorithm performs better than the HK-relax algorithm for detecting is a good cluster with a small conductance on communities and a collaboration network datasets but this takes more runtime for computation. After studying SimpleLocal algorithm with different locality parameter values, the good conductance cut is achieved by using a small value of locality δ near to 0 but using a large value near to 1, this gives a weak conductance cut. A Seed set size effect on HK-relax and SimpleLocal algorithms, with increasing the size of the seed set, a good conductance cut is obtained. HeatKernel algorithm is used for directed and undirected graphs, but SimpleLocal is used for undirected graph so the future work intends to solve this.

VII. REFERENCES

[1] W. Fan, and A. Bifet, "Mining Big Data: Current Status, and Forecast to the Future," SIGKDD Explorations Newsletter, vol. 14(2), pp. 1–5, 2013.

[2] U. Kang and C. Faloutsos, "Big graph mining: Algorithms and discoveries," SIGKDD Explorations, vol. 14 (2), 2012.

[3] S. Aridhiand E. M. Nguifo, "Big graph mining: Frameworks and techniques," Big Data Research, Vol. 6, pp. 1-10, 2016.

[4] S. E. Schaeffer, "Graph clustering," Computer science review, vol. 1(1), pp. 27-64, 2007.

[5] C. C. Aggarwal, and H. Wang, "A survey of clustering algorithms for graph data," Managing and mining graph data, vol. 40, 275-301, 2010.

[6] J. Ni, H. Fei, W. Fan, and X. Zhang, "Cross-network clustering and cluster ranking for medical diagnosis," In: ICDE, 2017.

[7] J. Ni, M. Koyuturk, H. Tong, J. Haines, X. Rong, and X. Zhang, "Disease gene prioritization by integrating tissue-specific molecular networks using a robust multinetwork model," BMC Bioinform, vol. 17(1), 453, 2016.

[8] D. Zhou, and C.J.C. Burges, "Spectral clustering and transductive learning with multiple views," In: ICML,2007.

[9] A. Kumar, P. Rai, and H. Daume, "Co-regularized multi-view spectral clustering," In: Advances in neural information processing systems, 2011.

[10] A. Kumar, and H. Daume, "A co-training approach for multi-view spectral clustering," In: ICML. 2011.

[11] W. Cheng, X. Zhang, Z. Guo, W. Yubao, P.F. Sullivan, and W. Wang, "Flexible and robust co-regularized multi-domain graph clustering," In: KDD, 2013.

[12] J. Ni, H. Tong, W. Fan, and X. Zhang, "Flexible and robust multi-network clustering," In: KDD, 2015.

[13] K. Kloster, and D. F. Gleich, "Heat kernel based community detection," Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014.

[14] N. Veldt, D. F. Gleich, and M. W. Mahoney, "A simple and strongly-local flow-based method for cut improvement," InternationalConference on Machine Learning (ICML), 2016.

[15] D. A. Spielman, and S.-H. Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, ACM, 2004.

[16] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," Proceedingsof 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), 2006.

[17] D. A. Spielman, and S.-H. Teng. "A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning," Society for Industrial and Applied Mathematics, vol. 42(1), pp. 1–26, 2013.

[18] K. Fountoulakis, X. Cheng, J. Shun, F. R. Khorasani, M. W. Mahoney, "Exploiting optimization for local graph clustering," arXiv preprint arXiv:1602.01886, 2016.

[19] R. Andersen, and K. J. Lang."An algorithm for improving graph partitions," Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2008.

[20] R. Andersen, and Y. Peres. "Finding sparse cuts locally using evolving sets," Proceedings of the forty-first annual ACM symposium on Theory of computing, ACM, 2009.

[21] T. C. Kwok, and L. C. Lau, "Finding small sparse cuts by random walk," Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Springer,pp.615-626, 2012.

[22] K. George, and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," SIAM Journal on Scientific Computing, vol. 20(1), pp. 359–392, 1998.

[23] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts." In SIGKDD'04, pp. 551–556. ACM, 2004

[24] S. v. Dongen, "Graph clustering by flow simulation," PhD Thesis, 2000.

[25] F. Chung, "A local graph partitioning algorithm using heat kernel pagerank," Internet Mathematics vol. 6(3), pp. 315-330, 2009.

[26] L. Orecchia, and Z. A. Zhu, "Flow-based algorithms for local graph clustering," Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, pp. 1267–1286, 2014.

[27] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich "Local Higher-Order Graph Clustering," Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 555-564, 2017.

[28] Dataset link: https://sparse.tamu.edu/.