International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-3, Issue-6, June 2015

Detecting Community Structures in Signed Social Networks (An Automated Approach)

Himanshu Sharma, Vishal Srivastava

***Abstract*— Many complex systems in the real world can be modeled as signed social networks. Community detection in signed social networks is a challenging research problem aiming at finding groups of entities having positive connections within the same cluster and negative relationships between different clusters. Many community detecting algorithms have been developed in the past. But, most of them are only effective for networks containing only positive relations and, are not suitable for signed social networks.**

This work is primarily for the networks having both positive and negative relations; these networks are known as signed social network. In this work DFA (detection and formation algorithm) has been proposed which works in two phases. The first phase is based on Breadth First Search algorithm which makes community structure on the basis of the positive links only. The second phase takes the output of first phase as its input and produces community structure on the basis of a robust criteria termed as participation level. Proposed algorithm can find the signed social networks where the negative inter-community links and the positive intra-community links are dense. Proposed algorithm is also useful in detecting the communities from only positive conventional graphs. Moreover it doesn't require any external parameter for its operation as is the case with other algorithms like FEC (finding and extracting communities). Inclusion of a new node in the graph is tackled effectively to reduce the unnecessary computation. This algorithm proceeds in breadth first way and incrementally extracts communities from the network. This algorithm is simple, fast and can be scaled up easily for large social networks.

The effectiveness of this approach has been demonstrated through a set of rigorous experiments involving both benchmark and randomly generated unsigned and signed networks. The algorithm is simulated by using GUESS (Graph Exploration System) tool. Results provided by proposed algorithm are good and comparable with other algorithms for unsigned and signed social networks in terms of accuracy and order of time complexity.

***Index Terms*—Community Detection, Community Structure Social Networks, Signed Social Networks,**

I. Introduction

Social networks are formed by individuals having some properties in common. A social network can be defined as a graph G = (V, E), where V = (v_1,v_2,v_3, . . . v_n) is the set of vertices, and E = (e_1,e_2,e_3, … e_n) is the set of edges connecting pairs of vertices. For example, in a human social network, each vertex (node) denotes an individual, and each edge (link) denotes a relation between two nodes. In weighted social networks, each link is attached with a real number called weight which represents in some sense how closely connected the vertices are [7]. In the field of social science, the networks that include only positive links are also called positive social networks, and the networks with both positive and negative links are called signed social networks [3] or signed networks for short.

Himanshu Sharma, Computer Science & Engineering Department, Arya College of Engineering & Information Technology, Jaipur, Rajasthan, India, 9460867761.

Vishal Srivastava, Computer Science & Engineering Department, Arya College of Engineering & Information Technology, Jaipur, Rajasthan, India, 9214052387.

A signed social network in its simplest form can be viewed as a weighted bidirectional graph having three types of weights {+1,0,-1} [3]. Weight "+1" is assigned to the edges connecting positively a pair of nodes, Weight "-1" is assigned to the edges connecting negatively a pair of nodes and Weight "0" is assigned if an edge does not exist between the nodes. For example, a network of nations where positive relation shows the political alliance and negative shows the opposition. In the friends-enemies network, positive link shows that they are friends and the negative shows that they are enemies.

In the literature, there are a number of examples of weighted graphs in which the weights assigned to the edges lies in a particular range of numbers. However, these graphs may be considered as a special case of the previously explained signed graph, we can transform these types of graphs to simple signed graphs by assigning +1 to the weights above a predefined threshold and -1 to the weights less than that level. This generalization of social networks is done because normally it is not easy or fair to give weights to the relationship of an individual with other individuals.

II. Previous Work on Detecting Community Structures in Social Networks

In context of social networks the task of grouping the set of vertices exhibiting similar properties or behavior is referred as Community Detection. Social networks are generally sparse in global yet dense in local. They have vertices in a group structure such that the vertices within the groups have higher density of edges while vertices among groups have lower density of edges. This kind of structure is called the *community* which is an important network property and can reveal many hidden features of the given networks. Two vertices having the same attribute have a positive link between them and the vertices having the opposite attribute will have a negative link. These vertices are classified on the basis of both the link density and the signs of the link. This task becomes challenging when there are some negative links within group and at the same time, some positive links between groups.

In the literature, many algorithms have been proposed to detect network communities or sub-graph clustering only in positive networks. They may be categorize into three groups as follows [1]: (1) Graph theoretic methods like Random walk methods, physics-based methods, and Spectral methods (2) Divisive algorithms like 'Betweenness' algorithms of Girvan and Newman [7] Tyler algorithm [8], and Radicchi algorithm [9] in which they divide the network into smaller subsections (3) Agglomerative algorithms like Modularity based algorithms [10],[26] which form communities by joining nodes together. Girvan and Newman [7] 'Betweenness' measure iteratively removes edges with the highest "stress" to eventually find disjoint communities. Clauset [10] suggested a faster algorithm but the number of clusters must still be specified by the user. Flake et al. [11] used the max-flow min-cut formulation to find communities around a seed node; however, the selection of seed nodes is not fully automatic. Kelsic [12] proposed an agglomerative algorithm for constructing overlapping communities using local shells, and implement methods for visualizing overlap between communities. Pons and Latapy [13] proposed a community finding algorithm based on random walk. This random walk starts from a single node treating it as a community and repeatedly performs the merging of a pair of adjacent communities that minimizes the mean of the squared distances between each node and its community. Hildrum [14] presented a cut-based focused community search algorithm. Palla [15] used clique percolation for the problem of identifying communities, where one node can belong to more than one community. Their method first identifies allcliques of the network and performs a standard component analysis of the clique-clique overlap matrix to discover a set of k-clique-communities. M.P.S Bhatia [22] recently introduced BFC (breadth first clustering) algorithm in which the communities are formed by clustering groups of nodes closely connected to each other.

The algorithm uses breadth-first traversal, as discussed in Cormen et al. [23], as its propagation method. With every vertex traversed, visit counter of all its neighbors is incremented and the vertices are enqueued in the Queue as used in breadth first search. The next vertex to be traversed is the vertex at of the front end. When the algorithm reaches on a vertex having visit counter greater than 2, it signifies that a cluster may exist. If neighbors of a vertex belong to more than one class then the vertex is assigned to the class with maximum common class neighbors.

Yang et al. [3] introduced a new algorithm FEC which works on both parameters i.e. on both link density and sign of the link. The main idea behind the algorithm is an agent-based random walk model, based on which the FC phase can find the sink community. This sink community is extracted from the entire network by the EC phase based on some robust graph cut criteria. In find community (FC) phase a sink node is placed by agent and calculates l-step transfer probability distribution function for each node. The l-step transfer probabilities are then sorted to find the nodes with least probabilities. The nodes with least l-step transfer probabilities represent the nodes outer to community and thus remove them to find the community structure.

III. COMMUNITY DETECTION ALGORITHM

A. *The main idea*

The communities in a network are formed by clustering groups of nodes closely connected to each other. The algorithm uses breadth-first traversal, as discussed in Cormen *et al.* [23], as its propagation method.

Proposed algorithm works in two phases:

PHASE 1 (Ignores negative edges).

The first phase, is only concerned with the positive links present in the graph. In this phase a breadth first traversal will be initiated from a particular vertex, which can be chosen randomly, and will proceed to traverse all its neighboring vertices. This approach will be advantageous in a way that the vertices which can form a community are traversed first then vertices of other community. Whenever a vertex is traversed all its neighbor's visit counter is incremented, when we reach on a vertex having visit counter 2 or more, it signifies that a cluster may exists if majority of its neighbors are traversed more than twice. See Fig.1.

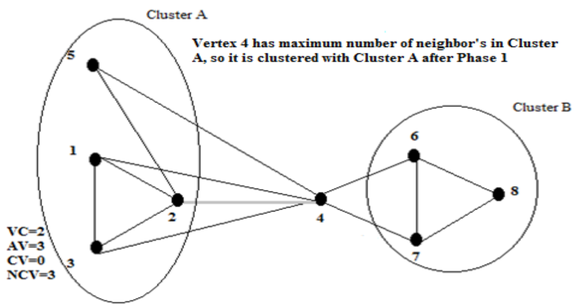

Fig.1(a) Shows parameter values with respect to the Vertex 3 and Maximum Participating Cluster for Vertex 4.

Small tightly coupled components are detected first which merges nearby vertices together to form larger cluster on the basis of majority of participation incrementally. There are cases when the vertex belong to more than one cluster then the vertex is assigned to the cluster in which it has maximum number of neighbors that is to maximum participation cluster as shown in the Fig.1(a). If vertex has equal participation in the clusters then it can be assigned to the any cluster, normally in this situation it is clustered with the cluster in which it is grouped first.

Every vertex V has four parameters:

1. AV : Number of adjacent vertices of V
2. NCV: No. of Non Clustered vertices adjacent to V having visit counter > = 2
3. CV : Number of clustered vertices adjacent to V having visit counter >=2
4. MPC : Majority participation cluster No.

In the above example vertex 4 has total five neighbours, from which 3 neighbors belong to cluster A (1,3,5) and 2 neighbors belong to cluster B (6,7) . Edge joining vertex 4 to vertex 2 is negative edge and according to the algorithm is neglected during the Phase1. They are not even counted in AV. Here vertex 4 has maximum participation in clusters A, so vertex 4 will be merged in cluster A.

PHASE 2

The output of the first phase, i.e., the clustered graph, which contains the knowledge of every cluster formed and the containing vertices. The main idea behind this phase is to reclassify the vertices with the negative edge on the basis of the participation level of the vertex having the negative edge, which can be defined as follows:

Participation level of vertex V_P = ((Total no. of +ve edges within the cluster Ci) / (Total no. of edges within the same cluster))

Where i=1,2,3,4,.. N; N= total no. of clusters formed.

The value of V_P lies between 0 and 1. When the node doesn't have any negative edge to other node within the cluster the value of V_P will always be the maximum i.e. 1. The cluster having the highest participation level will be awarded with the vertex.

Table I : Shows Participation level for the negative edge vertices of Fig. 1(a).

Vertex	**Cluster (No. of positive edges with in the cluster, V_P)**	**V_P Max.**
V_2	A(3, 3/4), B(0,0)	3/4
V_4	A(3, 3/4), B(2, 1)	1

Table I shows that V_2 has a maximum V_P in cluster A. So it remains in cluster A.

Vertex V_4 has a maximum V_P in cluster B, So according to the algorithm now in this phase this vertex will be re-clustered and it will break its association with the previous cluster i.e. cluster A and will join cluster B as shown in Fig. 1(b). This is same as in real life where a person wants to join a group which has 2 friends rather than the group which has 2 friends and an enemy.

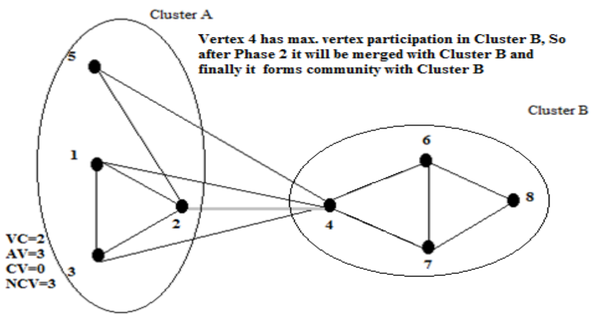

Fig.1(b) Shows the final cluster formed after Phase 2.

B. *The Algorithm*

Following is pseudo code for the algorithm. The algorithm uses queue data structure is represented by Q1and Q2 having enqueue and dequeue operations.

Phase 1 //treating the graph on the basis of the positive edges only

```
DFA(G, U) //U is the initial vertex
struct cluster_info{cluster name, size} ,
int tnn=total no. of nodes in graph
For every vertex having positive edge,
  begin1
    Enqueue(Q1, U)
    set U as visited
    while Q is not empty
      begin2
      H ← Dequeue(Q1)
      for each N ∈ Neighbors(h) in G
        begin3
        Increment VisitCounter(N)
        if N is not-visited
          begin4
          enqueue(Q1, N)
          set N as visited
          Sort (Q1) in decreasing order of visit
            counters by using insertion sort
            End 4
if VisitCounter(H) > 1
  begin5
  :label1 if (CV+NCV) ≥ Ceiling(AV/2)
    Begin6
      if NCV >CV
      begin7
         form set Sucv of Un-Clustered vertices
         \\ New Class Formed set class(Sucv) =C+1
      end7
      else if CV>NCV
      begin8
      Find MPC \\ like in Fig.1
      \\ merged into class set class(H) =MPC
      End8
    End6
  End5
End 3
End2
For all vertices left Un-Clustered
Put them in their MPC
If tnn > sum of total no. of clustered nodes //a new node
found in the graph
Begin9
Goto label1 with un-clustered node as parameter
End9
Return CG // clustered graph
End 1
```

```
Phase2(CG) // clustered graph CG passed as parameter

Begin1
  For every cluster Ci ,i={1,2,3… n} ,in the graph
    Begin2
      Find the vertex Vi with –ve edge
      ENQUEUE ((vi ,Q2)
    End2
  DEQUEUE(Vi , Q2)
    Begin 3,
      For clusters C1 , C2 , ……..Cn ,
      Find the no. of +ve edges, PE, with which the
      node is joined in the cluster
      Arrange the clusters in the decreasing order of the
      participation of that particular node.
    End 3
    Begin4
      For each cluster C1 ,C2 …Cn if PE > 1
      Find participation level for Vi ,VPi =((Total no.
```

of +ve edges within the cluster, P_E) / (Total no. of edges with in the cluster))
End4
Find the cluster C_{MP} with the maximum participation level
If C_{MP} is not same as the previous cluster
Remove the vertex from the previous cluster and add it to C_{MP}
End1

C. Evaluation of Algorithm

A signed social network example with 36 nodes having total 74 edges out of which 5 edges are negative is shown in Fig 2. The ovals shown in the figure denotes the communities formed by the each phase of the algorithm**.** Traversing of graph can be started from the randomly chosen node 0. The proceedings of the first phase of algorithm are shown in Table II. The column 2 of Table II shows the vertex under consideration and in "( )" its visit counter is shown. Column 3 shows the adjacent vertex of V and their visit counters. Column 4 shows the current status of the vertex queue which is been updated in the whole algorithm. This queue stores all the unvisited nodes which have been traversed at least once. The vertex at the front end of the queue is the next vertex to be evaluated.

This vertex queue is maintained in the increasing order of the visit counters of the participant vertices. The proceedings are shown in the following Table II, where C denotes the cluster formed. The cluster name followed by " ' " shows the modified clusters formed during the phase. Now V_P is calculated for V_4. V_4 has 2 positive links with the cluster A. The total number of edges linked with V_4 is 3. So according to our previously stated formula the value of V_{P4} comes out to be 2/3 for cluster A, and similarly it is calculated for cluster B and cluster C and is written in the fourth column of the Table III. After the calculation of V_P in this phase only change that can be seen in the previous clustered graph by phase 1, shown in dotted ovals, is that the node V_4 which was the part of the cluster A is now reassigned to the cluster C because the participation level of the vertex is maximum in this cluster i.e. 1. All other clusters remains unchanged.

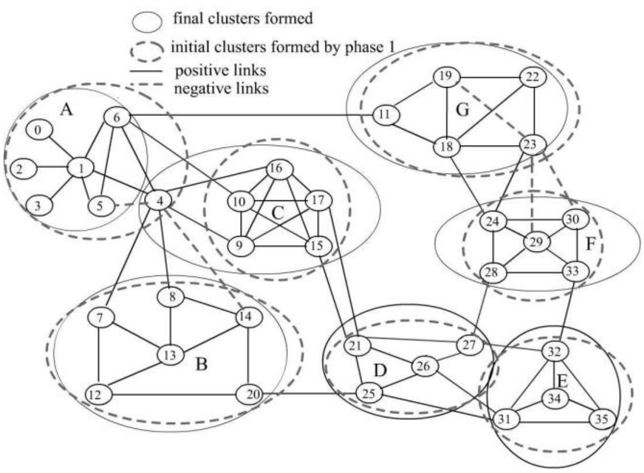

Fig. 2 A Signed Social Network Example.

Table II : Shows the step by step proceedings of Phase 1 on the Signed Social Network shown in Fig. 2.

Steps	V(VC)	Adjacent vertices of V(VC)	Sorted vertex Queue (Q1)	C
1	0(0)	1(1)	1(1)	
2	1(1)	0(1),2(1),3(1),4(1),5(1) 6(1)	2(1),3(1),4(1), 5(1),6(1)	
3	2(1)	1(2)	3(1),4(1),5(1),6(1)	
4	3(1)	1(3)	4(1),5(1),6(1)	
5	4(1)	1(4),6(2),7(1) 8(1),9(1),16(1)	6(2),5(1),7(1),8(1),9(1),16(1)	
6	6(2)	1(5),4(2),5(2), 10(1),11(1)	5(2),7(1),8(1),9(1),16(1), 10(1),11(1)	A
7	5(2)	1(6),6(3)	7(1),8(1),9(1),16(1),10(1), 11(1)	
8	7(1)	4(3),12(1),13(1)	8(1),9(1),16(1),10(1),11(1) ,12(1),13(1)	
9	8(1)	4(4),13(2),14(1)	13(2),9(1),16(1),10(1),11(1),12(1),13(1),14(1)	
10	13(2)	7(2),8(2),12(2), 14(2)	12(2),14(2),9(1),16(1),10(1),11(1)	B
11	12(2)	7(3),13(3),20(1)	14(2),9(1),16(1), 10(1),11(1), 20(1)	
12	14(2)	8(3),13(4),20(2)	20(2),9(1),16(1), 10(1),11(1)	
13	20(2)	12(3),14(3),25(1)	9(1),16(1),10(1),11(1),25(1)	B'
14	9(1)	4(5),10(2),15(1), 16(2),17(1)	16(2),10(2),11(1),25(1),15(1),17(1)	
15	16(2)	4(6),10(3),9(2), 15(2),17(2)	10(3),15(2),17(2),11(1),25(1)	C
16	10(3)	6(4),9(3),15(3), 16(3),17(3)	15(3),17(3),11(1),25(1)	
17	15(3)	9(4),10(4),16(4),17(4), 21(1)	17(4),11(1),25(1),21(1)	
18	17(4)	9(5),10(5),16(5),15(4), 21(2)	21(2),11(1),25(1),	
19	21(2)	15(5),17(5),25(2),	25(2),11(1) ,26(1),27(1)	
20	25(2)	20(3),21(3),26(2), 31(1)	26(2),11(1),27(1),31(1)	D
21	26(2)	21(4),25(3),27(2), 31(2)	31(2),27(2),11(1)	
22	31(2)	25(4),26(3),32(1),34(1), 35(1)	27(2),11(1),32(1),34(1),35(1)	
23	27(2)	21(5),26(4),28(1), 32(2)	32(2),11(1),34(1),35(1),28(1)	D'
24	32(2)	27(3),31(3),33(1),34(2),35(2)	34(2),35(2),11(1), 28(1),33(1)	E
25	34(2)	31(4),32(3),35(3)	35(3),11(1), 28(1),33(1)	
26	35(3)	31(5),32(4),34(3)	11(1), 28(1),33(1)	
27	11(1)	6(5),18(1),19(1)	28(1),33(1),18(1),19(1)	
28	28(1)	27(4),33(2),29(1),24(1)	33(2),18(1),19(1),29(1),24(1)	
29	33(2)	30(1),32(5),28(2), 29(2)	29(2),18(1),19(1),24(1),30(1)	F
30	29(2)	30(2),33(3),28(3), 24(2)	30(2),24(2),18(1),19(1)	
31	30(2)	24(3),29(3),33(4)	24(3),18(1),19(1)	F'
32	24(3)	18(2),23(1),29(4), 28(4),30(3)	18(2),19(1),23(1)	F''
33	18(2)	11(2),19(2), 22(1),23(2) 24(4)	19(2),23(1),22(1)	G
34	19(2)	11(3),18(3),22(2)	23(2),22(2)	
35	23(2)	24(5),18(4),22(3)	22(3)	
36	22(3)	19(3),18(5),23(3)		G'

New cluster formed in column 2 of Table II .

Merged with its MPC

Unclustered node with degree ≥ 2

Table III : Shows Participation level for the negative edge vertices of Fig. 2.

Vertex Queue Q2 : 4,5,14,19,23,29,30				
Steps	Vertex	Cluster(No. of +ve edge within)	VP	C_{MP}
1	4	A(2)	2/3	C
2		B(2)	2/3	
3		C(2)	1	
4	5	A(2)	2/3	A
5	14	A(0)	0	B
6		B(2)	1	
7	19	G(3)	3/4	G
8	23	G(2)	2/3	G
9		F(1)	1/3	
10	29	F(4)	1	F
11	30	F(3)	1	F

The proposed algorithm is simulated using Guess software and applied on various bench marked examples of signed and unsigned networks like Gahuku-Gama Subtribes Network[5] and Zachary Karate Club network [29]. The proposed algorithm detected the communities in these bench marked networks successfully. The communities detected in Gahuku-Gama Subtribes Network is shown in Fig. 3(b) which are same as detected by Bo Yang et.al [3].

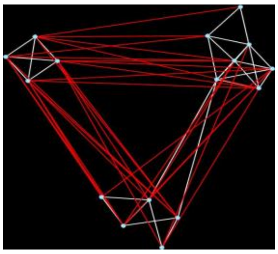

Fig. 3(a) Gahuku -Gamma subtribes network

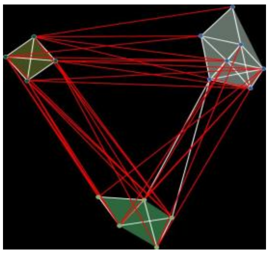

Fig. 3(b) Communities detected by proposed algorithm

The proposed algorithm has been applied on various sized networks and the average execution time is shown in Fig. 4 The outcome shows its effectiveness and nearly linear execution time with growth in size of network.

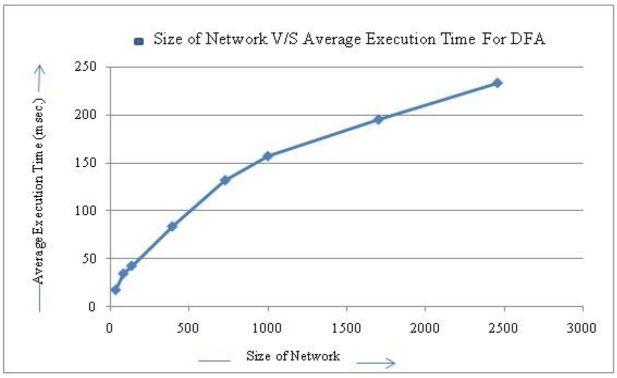

Fig. 4 Average Execution time of proposed algorithm for various Size of Networks

IV. CONCLUSION AND FUTURE WORK

The paper proposed a simple approach which can detect communities in both signed and unsigned social networks. Proposed algorithm considers both the link density and signs for mining signed network community and is automatic i.e. it doesn't requires any external parameter as with the case of FEC algorithm [3]. Real world social networks are subject to lot of change with time, so there are always new nodes that join the graph frequently. In proposed algorithm this problem is effectively tackled due to which it is not required to run the whole algorithm again even when the changes in the structure have been brought by only one new node.

The time complexity of the algorithm is O(V+E) where V represent number of vertices and E represent edges in the network. This algorithm is simple, fast and can be scaled for large social networks. The effectiveness of this approach has been validated using bench marked network examples. So far proposed algorithm is tested with medium sized networks, in the future it will be enhanced to deal with large and dynamic networks of order higher than 10^5.

V. REFERENCES

[1] S.H. Strogatz, "Exploring Complex Networks," Nature, Vol. 410, 268-276, March 2001.
[2] D.J. Watts and S.H. Strogatz, "Collective Dynamics of Small World Networks," Nature, Vol. 393, 440-442, June 1998.
[3] Bo Yang, W.K. Cheung, and Jiming Liu, "Community Mining from Signed Social Networks," IEEE / Knowledge and Data Engineering, Vol. 19, No. 10, 1333-1348, October 2007.
[4] P. Doreian and A. Mrvar, "A Partitioning Approach to Structural Balance," Social Networks, Vol. 18, No. 2, 149-168, 1996.
[5] K.E. Read, "Cultures of the Central Highlands, New Guinea," Southwestern J. Anthropology, Vol. 10, No. 1, 1-43, 1954.
[6] N. Bansal, A. Blum, and S. Chawla, "Correlation Clustering," Proc.43rd Ann. IEEE Symp. Foundations of Computer Science (FOCS '02), 238-247, 2002.
[7] M. Girvan and M.E.J. Newman, "Community Structure in Social and Biological Networks," Proceedings of the National Academy of Sciences, Vol. 99, No. 12, 7821-7826, June 2002.
[8] J.R. Tyler, D.M. Wilkinson, and B.A. Huberman, "Email as Spectroscopy: Automated Discovery of Community Structure within Organizations," *Proc. First Int'l Conf. Communities and Technologies (C&T '03),* 81-96, 2003.
[9] F.Radicchi and C.Castellano, "Defining and Identifying Communities in Networks," Proceedings of the National Academy of Sciences, Vol. 101, No. 9, 2658-2663, March 2004.
[10] Clauset, M. E. J Newman and C. Moore, "Finding community structure in very large networks", Physical Phys. Rev. E, Vol. 70, No. 6, 70(066111), 2004.
[11] G. W. Flake, S. Lawrence and C. Lee Giles, "Efficient Identification of Web Communities," ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining, Boston, Massachusetts, United States, 150 - 160, 2000.
[12] Eric D. Kelsic, "Understanding complex networks with community finding algorithms," SURF 2005.
[13] P. Pons and M. Latapy, "Computing Communities in Large Networks Using Random Walks," Proc. 20th International Symp. Computer and Information Sciences (ISCIS '05), 284-293, 2005.
[14] Kirsten Hildrum and Philip S. Yu, "Focused Community Discovery," 641-644, Fifth IEEE International Conference on Data Mining, ICDM, 2005.
[15] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society," Nature, Vol. 435, No. 7043, 814-818, 9 June 2005.
[16] D.H. Kim and H. Jeong, "Systematic Analysis of Group Identification in Stock Markets," Physical Rev. E, Vol. 72, 046133, 2005.
[17] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," European Physics Journal, 0605087v3, 2006.
[18] Rong Qian, Wei Zhang and Bingru Yang, "Detect community structure from the Enron Email Corpus Based on Link Mining," Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), 2006.
[19] Ryutaro Ichise and Hideaki Takeda, "A mining method of communities keeping tacit knowledge," Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06), 2006.
[20] Bo Yang and D.Y. Liu, "Force Based Incremental Algorithm for mining community structure in Dynamic Network," J. Comp. Sci. & Tech., Vol. 21, No.3, 393-400, May 2006.
[21] Bo Yang and Jiming Liu, "An Autonomy Oriented Computing (AOC) Approach to Distributed Network Community Mining," First IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'07), 2007.
[22] M.P.S. Bhatia and Pankaj Gaur, "Statistical approach for community mining in social networks," IEEE /SOLI ,2008
[23] Thomas H. Cormen, Charles E. Leiserson and Ronald L.Rivest, "Introduction to algorithms," The MIT Press and McGraw-Hill Book Company, 2001
[24] J. A. Barnes, "Class and committees in a Norwegian island parish," Human Relations, Vol. 7, 39-58, 1954.
[25] http://en.wikipedia.org/wiki/Social_network_analysis_software.
[26] M. E. J. Newman and M. Girvan, "Fast algorithm for detecting community structure in networks," Physical Review E-69:026113, 2004.
[27] Jordi Duch and Alex Arenas, "Community detection in complex networks using Extremal Optimization," cond-mat/0501368 v1, 2005.
[28] http://graphexploration.cond.org/
[29] W. W.Zachary, *J. Anthropol.*, "An information flow model for conflict and fission in small groups", JOURNAL OF ANTHROPOLOGICAL RESEARCH Vol. 33, No. 4, 452-473, 1977

Himanshu Sharma recieved his B.E degree in computer science from University of Rajasthan in year 2006. He is pursuing M.Tech presently. His main research interest includes data mining, network analysis and knowledge engineering.

Vishal Srivastava recieved his B.Tech & M.Tech in degree computer science from RGPV college in 2004 and 2007 respectively and pursuing his PhD presently. He is working as professor in Arya College of Engineering and Information Technology, Jaipur, Rajasthan. He is member of CSI, IAE, UACEE, SCIEI, IJCTT, IJATER. He is also editorial board member and reviewer in IJACR, IJRET, IJCST, IJCT, IJAERD, IJATEE etc in area of CS and IT. His area of interest includes Networking and Data Mining. He has published 75 International papers and 24 National papers in international journals, conferences and national conferences.He also get grants from DST, Government of Rajasthan for his R&D and projects like SAMADHAN, Vaccination schedule alert system on mobile, Petition signing an android application.