2009 International Conference on Computer Engineering and Applications
IPCSIT vol.2 (2011) © (2011) IACSIT Press, Singapore

Community Mining in Signed Social Networks –An Automated Approach

Tushar Sharma[1], Ankit Charls[2], P.K. Singh[3]

Atal Bihari Vajpayee- Indian Institute of Information Technology and Management, Gwalior, India
[1]tushar.sharma.9@gmail.com, [2]ankitcharles@gmail.com, [3]pksingh@iiitm.ac.in

Abstract. A social network can be viewed as a complex interconnection of social entities. Mining a community is the task of grouping these social entities together on the basis of their linked pattern. A lot of research has been done on this subject but most of them were only concerned with the unsigned graph. Our work is primarily for the networks having both positive and negative relations; these networks are known as signed social network. In this work, we propose CRA (Clustering re-clustering algorithm) which works in two phases. The first phase is based on Breadth First Search algorithm which forms clusters on the basis of the positive links only. The second phase takes the output of first phase as its input and produces clusters on the basis of a robust criteria termed as participation level. Our algorithm can mine the signed social networks where the negative inter-community links and the positive intra-community links are dense. The algorithm is also useful in mining the communities from positive only conventional graphs. Moreover it doesn't require any external parameter for its operation as is the case with other algorithms like FEC. Inclusion of a new node in the graph is tackled effectively to reduce the unnecessary computation.

Keywords: Social network; community mining; signed social network; Participation level

1. Introduction

The term social networking was first coined in 1954 by J.A Barnes [4]. A social community can be formed on web by the people sharing hobbies, working together, living together or having similar ideas about a subject. When we model the structure of the number of communities they form a complex network called a social network which represents a real world [2][3]. In mathematical terms this network can be best represented as G = (V, E), where graph G is a finite set of vertices V = (V1, V2, V3,…, Vn) and edges E is a finite set of edges connecting these vertices. Typically these graphs are very large and both the edges and the vertices have attributes. Examples of social networks include the Gahuku-Gama sub-tribes network [6] and the Zachary's Karate club network [18].

A signed social network in its simplest form can be viewed as a weighted bidirectional graph having two types of weights i.e. positive and negative weights [5]. Positive weights are represented as +1 and negative weights are represented as -1. For example, a network of nations where positive relation shows the political alliance and the negative shows the opposition. In the literature, we find a number 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 as 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.

2. PREVIOUS WORK ON COMMUNITY MINING

Community mining is a technique to classify the network nodes in a group or in a community within which the attribute of these nodes is maximized. 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 nodes 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] 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 all cliques 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 [16] 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. [17], as its propagation method. With every node traversed, visit counter of all its neighbors is incremented and the nodes are enqueued in the Queue as used in breadth first search. The next node to be traversed is the node at of the front end. When the algorithm reaches on a node 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. [1] 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.

3. CRA (CLUSTERING RECLUSTERING ALGORITHM)

3.1. The Main Idea

This algorithm works in two phases.

PHASE 1 (Ignores negative edges).The first phase, Clustering phase, takes the Breadth first search [17] as its basic algorithm. The similar approach was also used by BFC algorithm [16]. In breadth first traversal all the neighbor nodes in the community are traversed first and then the nodes of other community.

The algorithm starts from a specified node which can be randomly chosen. The visit counter of all of its neighbors is incremented. When it reaches a node having VC (visit counter) greater than 2 signifies a cluster as shown in the following figure 1

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 class then the vertex is assigned to the class with maximum class neighbor as shown in the Fig 1. Every node 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. Use the “Header 2” style, shown above, for subheads.

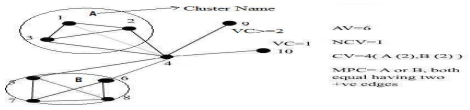

Fig. 1: Example showing all parameters with respect to node 4

In the above example four of the neighbors of vertex 4 belong to cluster A (3,2) and B (5,6) . Edges joining node 4 to node 1 is negative edge and according to the algorithm is neglected during the Phase1. They are not even counted in AV.

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 VP = ((Total no. of +ve edges within the cluster C_i) / (Total no. of edges within the same cluster)) Where i=1,2,3,4,.. N; N= total no. of clusters formed

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

Here in Table 1 we can see that V4 has a participation level of 1 in cluster B. So according to the algorithm now in this phase this node will be re-clustered and it will break its association with the previous cluster i.e. cluster A and will join cluster B. This is same as in real life we want to join a group which has 2 friends rather than the group which has 2 friends and enemy.

Here we can see that V_4 has a participation level of 1 in cluster B. So according to the algorithm now in this phase this node will be re-clustered and it will break its association with the previous cluster i.e. cluster A and will join cluster B. This is same as in real life we want to join a group which has 2 friends rather than the group which has 2 friends and 1 enemy.

Table 1: Participation level for the –ve nodes of fig: 1

Node	Cluster (no. of positive edges within the cluster , Vp)	Vp(max)
V_1	A (2 , 2/3)	2/3
V_4	A (2 , 2/3) , B(2 , 1)	1

3.2. The Algorithm

Following is pseudo code for the CRC algorithm. The algorithm uses queue data structure is represented by Q1 and Q2 having enqueue and dequeue operations.

```
Phase 1 //treating the graph on the basis of the positive edges only
BFC(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 ε 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
                                  vertices
 \\ New Class Formed set class(S_ucv) =C+1
                                  end7
                               else if CV>NCV
                                  begin8
                                    Find MPC  \\ like in Fig.2
 \\ 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 C_i ,i={1,2,3… n} ,in the graph
Begin2
Find the vertex V_i with –ve edge
ENQUEUE ((v_i ,Q2)
End2
DEQUEUE(V_i , Q2)
Begin 3,For clusters C_1 , C_2 , ……..C_n ,
Find the no. of +ve edges, C_N , 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 C_1 ,C_2 …C_n if C_N > 1
Find participation level for V_i ,$\mathbf{VP_i}$ =((Total no. of +ve edges within the cluster,C_n) / (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

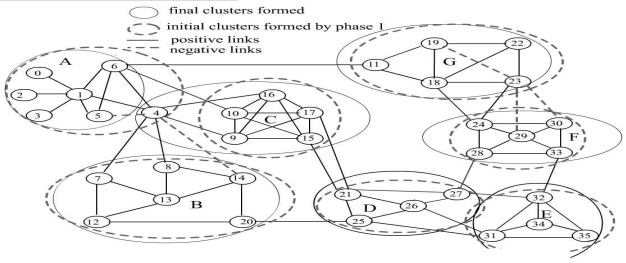

Fig. 2 : A network example

3.3. Evaluation of CRA

Here is a network example with 36 nodes having total 74 edges out of which 5 edges are negative, Fig 2. The ovals shown in the figure denotes the communities formed by the each phase of the algorithm**.** We traverse the algorithm starting from the randomly chosen node 0. The proceedings of the first phase of algorithm are shown in Table 2. The column 2 of Table 2 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 2, where C denotes the cluster formed. The cluster name followed by " **'** **"** shows the modified clusters formed during the phase.

Let's discuss how VP is calculated for V4. V4 has 2 positive links with the cluster A. The total number of edges linked with V4 is 3. So according to our previously stated formula the value of VP4 comes out to be 2/3 and is written in the final column of the table 2 .After the calculation of this phase only change that can b seen in the previous clustered graph by phase 1, shown in dotted ovals, is that the node V4 which was the part of the cluster A is been assigned to the cluster C because the participation level of the vertex is maximum in this cluster i.e. 1. All other clusters remain unchanged.

Table 2: Steps taken by the first phase of the algorithm

Step	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),1 7(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), 26(1),27(1)	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)	
Step	**V (VC)**	**Adjacent vertices of V(VC)**	**Sorted vertex Queue (Q1)**	**C**
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 1.
Merged with its MPC
Unclustered node with degree≥ 2

4. Conclusion and Future Work

Table 3: Steps taken by the Second Phase of the algorithm

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

In this paper we present a simple robust algorithm which can find communities even in unpartitionable and unclusterable [1] signed social networks. Our 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 [1]. Also all real social networks are subject to lot of change with time. So there are always new nodes that join the graph frequently. Here in our algorithm we propose a method by 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. Our work is under progress and proposes a method of efficient identification of the social network communities. We are currently working on formalizing the algorithm and developing a simulation and evaluation plan which will verify our work on the standard graph examples present in the literature.

5. References

[1] Bo Yang, W.K. Cheung, and Jiming Liu, "Community Mining from Signed Social Networks", IEEE / KDE, VOL. 19 pp 1333-1348, NO. 10,2007.

[2] S.H. Strogatz, "Exploring Complex Networks," Nature, vol. 410, pp. 268-276, 2001.

[3] D.J. Watts and S.H. Strogatz, "Collective Dynamics of Small-World Networks,"Nature, vol.393, pp.440-442,1998.

[4] J. A. Barnes, "Class and committees in a norwegian island parish," Human Relations, 1954.

[5] P. Doreian and A. Mrvar, "A Partitioning Approach to Structural Balance," Social Networks, vol. 18, no. 2, pp. 149-168, 1996.

[6] K.E. Read, "Cultures of the Central Highlands, New Guinea," S.W J. Anthropology, vol. 10, no. 1, pp. 1-43, 1954.

[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, pp. 7821-7826, 2002.

[8] J.R. Tyler, D.M. Wilkinson, and B.A. Huberman, "Email as Spectroscopy: Automated Discovery of Community Structure within Organizations", C&T '03, pp. 81-96,2003.

[9] F.Radicchi, C.Castellano, "Defining and Identifying Communities in Networks," Proceedings of the National Academy of Sciences, vol. 101, no. 9, pp. 2658-2663, 2004.

[10] A. Clauset, M. E. J Newman & Moore, C. "Finding community structure in very large networks", Physical Phys. Rev. E ,Volume 70 , Issue 6,70(066111), 2004.

[11] G. W. Flake, S. Lawrence, C. Lee Giles, "Efficient Identification of Web Communities", ACM SIGKDD, international conference on Knowledge discovery and data mining Boston, United States pp: 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 Int'l Symp. Computer and Information Sciences (ISCIS '05), pp. 284-293, 2005.

[14] Kirsten Hildrum, Philip S. Yu, "Focused Community Discovery", pp.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, pp. 814-818, 9 June 2005.

[16] Bhatia, M.P.S., Gaur, Pankaj "Statistical approach for community mining in social networks", IEEE /SOLI ,2008

[17] Thomas H. Cormen, Charles E. Leiserson and Ronald L. Rivest, Introduction to algorithms, The MIT Press and McGraw-Hill Book Company, 2001

[18] W.W. Zachary, "An Information Flow Model for Conflict and Fission in Small Groups," J. Anthropological Research, vol. 33,pp. 452-473, 1977.