SURVEY ON COMMUNITY DETECTION IN SOCIAL NETWORKS

Sneha Kamal[1], Reshmi.S[2]

[1]*Department of Computer Science and Engineering, SBCE, Pattoor (India)*

[2]*Assistant Professor, Department of Computer Science and Engineering, SBCE, Pattoor (India)*

ABSTRACT

Community detection is a growing field in the area of Social Network applications. Community in social network is peoples with common interests join anytime anywhere to freely share information, experiences, opinions, services, and other useful resources. Communities give us valuable data about relation between individuals and how data transfers between them. . Social network mainly have a graphical representation. Within the graph community is a group of vertices which probably share common properties and play similar roles. A few methods focus here to detect communities are by analysing the graphical structure of the network, and by applying clustering techniques on network. This will identify the disjoint communities in a network.

***Keywords**: Affinity Propagation, Clustering, Gibbs sampling, Hierarchical Agglomeration, Social Network.*

I. INTRODUCTION

Many systems of current interest can usefully be represented as networks. Examples include the Internet and the world-wide web, social networks [12] food webs and biochemical networks. A network community generally refers to a group of vertices within which the connecting links are dense. The links represent the data connections between computers, friendships between peoples and so forth. Interactions between nodes can be used to determine communities in a social network. Particularly, network communities in different contexts may be circles of a society within which people share common interests and keep more contacts, clusters of web pages related to common topics.

A community is formed by individuals such that those within a group interact with each other more frequently than with those outside the group. Therefore, communities are groups of entities that presumably share some common properties. Community detection is important for many reasons, including node classification which entails homogeneous groups, group leaders or crucial group connectors. Communities may correspond to groups of pages of the World Wide Web dealing with related topics to functional modules such as cycles and pathways in metabolic networks to groups of related individuals in social networks and so on. Community detection is discovering groups in a network where individuals' group memberships are not explicitly given.

Human beings are social. The Easy way of using social media allows people to extend their social life in unprecedented ways. Difficult to meet friends in the physical world, but it is much easier to find friend through online with similar interests.

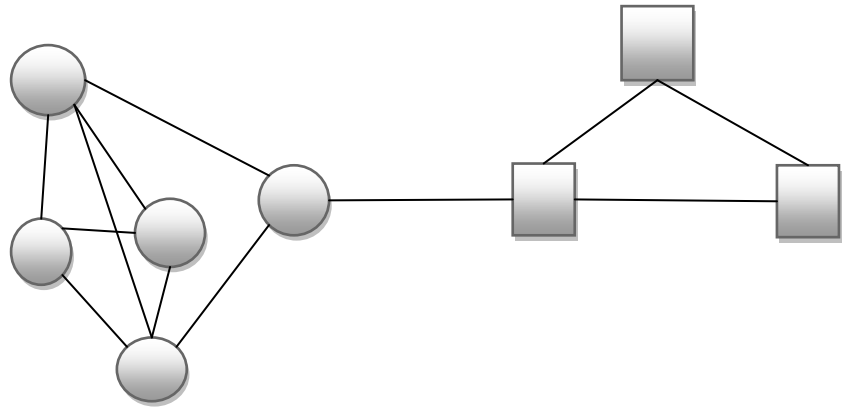

Fig1: A Social Network with Two Communities

II. FINDING COMMUNITIES IN A NETWORK

2.1 Semantic Community Discovery

A semantic [3] community includes users with similar communication interests and topics that are associated with their communications. The community structure of an SN is studied by modelling the communication documents exchange between the users. Communication documents include emails. Because email body contains valuable information regarding shared knowledge and the SN infrastructure. A Bayesian network is used to simulate the generation of emails in SNs. Differing in weighting the impact of a community on users and topics, two versions of Community models are proposed.

2.1.1 CUM: Modeling community with users

Initially consider a Social Network (SN) community as no more than a group of users. This is similar to the assumption in a topology-based method. For a specific topology-based graph partitioning algorithm such as Modularity, the connection between two users can be simply weighted by the frequency of their communications. In Community User Model (CUM) treat each community as a multinomial distribution over users. Each user u is associated with a conditional probability P (u/c) which measures the degree that u belongs to community c. The goal is therefore to find out the conditional probability of a user given each community. Then users can be tagged with a set of topics, each of which is a distribution over words. Community discovered by CUM is similar to the topology-based algorithm.

2.1.2 CTM: Modeling community with topics

In contrast to CUM, second model introduces the notion that an SN community consists of a set of topics, which are of concern to respective user groups. Analogously, the products of Community Topic Model (CTM) are a set of conditional probability P(z/c) that determines which of the topics are most likely to be discussed in community c. Given a topic group that c associates for each topic z, the users who refer to z can be discovered by measuring P(u/z).

CTM differs from CUM in strengthen the relation between community and topic. In CTM, semantics play a more important role in the discovery of communities. Similar to CUM, the side effect of advancing topic z in the generative process might lead to loose ties between community and users. An obvious phenomena of using CTM is that some users are grouped to the same community when they share common topics even if they

correspond rarely. For CUM, users often tend to be grouped to the same communities while CTM accentuate the topic similarities between users even if their communication seems less frequent.

2.1.3 The Algorithms

Gibbs sampling for Community-User-Topic model

```
/* Initialization */
for each email d
  for each word wi in d
    assign wi to random community, topic and user;
    /* user in the list observed from d */
/* Markov chain convergence */
i ← 0;
I ← desired number of iterations;
while i < I
  for each email d
    for each wi ∈ d
      estimate P(ci, ui, zi|wi), u ∈ ad;
      (p; q; r) ← argmax (P (cp, uq, zr|wi));
      /*assign community p, user q, topic r to wi*/
      record assignment T (cp, uq, zr, wi);
    i + +;
```

2.2 Finding Community Structure in Very Large Networks

There are several methods to discover communities from social network. But most of the methods proposed so far are unsuitable for very large networks, because of their computational cost. Here consider a [4] hierarchical agglomeration algorithm for detecting community structure which is faster than many competing algorithms.

The algorithm modularity uses a greedy optimization in which, starting with each vertex being the sole member of a community of one, we repeatedly join together the two communities whose amalgamation produces the largest increase in members of community. For a network of n vertices, after n-1 such joins left with a single community and the algorithm stops.

The operation of the proposed algorithm involves finding the changes in Q that would result from the amalgamation of each pair of communities, choosing the largest of them, and performing the corresponding amalgamation. One way to implement this process is to think of network as a multigraph, in which a whole community is represented by a vertex, bundles of edges connect one vertex to another, and edges internal to communities are represented by self-edges. Rather than maintaining the adjacency matrix for calculating ΔQij, uses update a matrix. Since joining two communities with no edge between them can never produce an increase in Q. Only store ΔQij for those pairs i, j that are joined by one or more edges. Since this matrix has the same support as the adjacency matrix, it will be similarly sparse, so it can be represent with efficient data structures.

Maintain three data structures

- A sparse matrix containing ΔQij for each pair i,j of communities with at least one edge between them. We store each row of the matrix both as a balanced binary tree and max heep.
- A max-heap H containing the largest element of each row of the matrix ΔQij along with the labels i,j of the corresponding pair of communities.
- An ordinary vector array with elements a_i.

Where eij = 1/2m if i and j are connected and zero otherwise, and $a_i = k_i/2m$, m is the number of edges in the graph and degree k_i of a vertex i is the number of edges incident upon it. Thus we initially set

$$\Delta Qij = \begin{cases} 1/2m - k_i k_j /(2m)^2 & \text{if i, j are connected,} \quad (1) \\ 0 & \text{otherwise,} \end{cases}$$

and

$$a_i = k_i/2m \quad (2)$$

The Algorithm

1) Calculate the initial values of ΔQij and a_i according to (1) and (2), and populate the max-heap with the largest element of each row of the matrix ΔQ.
2) Select the largest ΔQij from H, join the corresponding communities, update the matrix ΔQ, the heap H and ai and increment Q by ΔQij.
3) Repeat step 2 until only one community remains.

The running time of the algorithm on a network with n vertices and m edges is O(mdlog n) where d is the depth of the dendrogram describing the community structure. Many real-world networks are sparse and hierarchical, with m ˜ n and d ˜ log n, in which case our algorithm runs in essentially linear time, O(n log2 n). This is considerably faster than most previous general algorithms, and allows us to extend community structure analysis to networks that had been considered too large to be tractable.

2.3 Community Detection Using Action of Users

The online social networks have a graph structures. It include effective information of users within networks. This information can lead to improve the quality of community discovery. Here [5] instead of using centrality measures in social networks analysis, use user actions to identify communities and leaders. First, based on Interests and activities of users in networks, discover some small communities of similar users, and then by using social relations, extend those communities. A Social graph is an undirected graph G = (V, E)

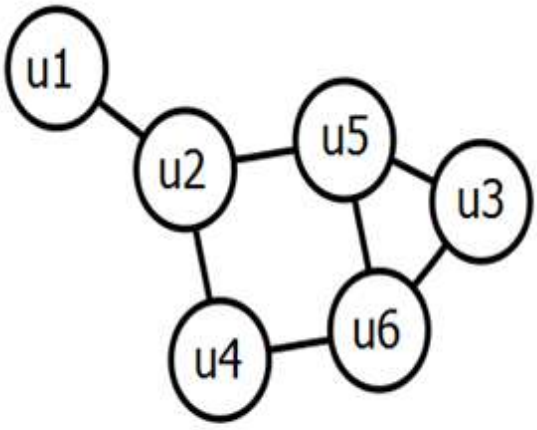

Fig.2: A Sample of Social Network

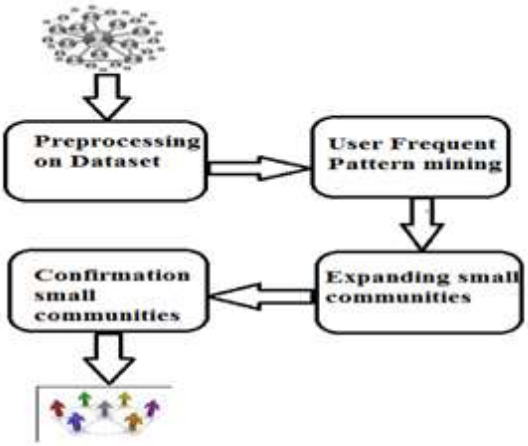

Fig.3: Steps of Community Detection

2.3.1 Pre-processing

The input of the algorithm includes user-action table and graphs. Two thresholds will be determined, β and ¥ that will be used in the algorithm.

Action	User
A1	U1,U3,U9,U2
A2	U1,U6,U7
A3	U2,U5,U8
A4	U2,U5
A5	U1,U6
A6	U4
A7	……..

TABLE I User-Action table

2.3.2 User Frequent Pattern mining

Find maximal patterns using frequent pattern mining algorithms and minimum support threshold ¥. For mining the patterns use Aprior pattern mining algorithm.

Frequent Pattern	Maximal Pattern
U1	U1,U6
U2	U2,U5
U5	
U1,U6	
U2,U5	

TABLE III If ¥=2, the maximal patterns include two tails

2.3.3 Confirmation Small Communities

Every small extracted group of previous step consists of few users. They are operationally similar to each other. To verify these are communities users of each group is connected to each other by a threshold. It is called β. In this step two or multiple groups are examined. If users within each group are connected according to a threshold they are refers to as a community. Otherwise group will be divided into small communities.

Steps involved in the confirmation of small communities

- Any node of each group stores its name or ID in its memory.
- All nodes which their memories are edited transmit their memories to neighbours.
- Neighbours integrate them to their memories.
- Variable called STEP which adds one unit of value to per sender node, maximum value is β.
- If all users of group were found in one memory, algorithm ends and return TRUE.
- Otherwise, value STEP is examined and if it is equivalent to value β, algorithm ends and return FALSE

2.3.4 Expanding Small Communities

Each extracted community in the previous step consists of two or more nodes with similar action which are to some extent related, that build a small community. Since users let their network friends see their actions, and seeing actions performed by their friends may sometimes tempt some fraction of the users to perform those actions themselves. So users who are in neighbourhood probably are more similar. It is sensible, due to

threshold β, these communities to be expanded. For achieving this goal, an algorithm similar to k-nearest neighbour, was used. Voting among neighbours is done to specify the node to communities.

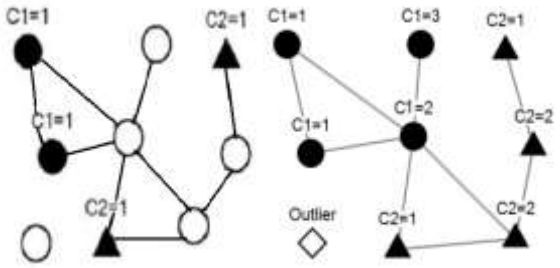

Fig.4: Triangle Community Scoring 6 and Circle Community Scoring 7

Compared with other methods, extracted nodes in the communities may not have the best density. But each community is expanded around the larger core of similar and related nodes. So in applications like predicting the users actions in marketing and recommender systems, the method will be more effective.

2.4 Community Discovering using Clustering Technique

A method is proposed [6] to derive communities of social networks. In addition to finding topics of network, this method also parses and analysis amount of communication between users. The method focus on text based information of social networks.

Various step involved are Social network Dataset Preparation module, Text Pre-processing and Data Modeling module, Social Object Clustering module, Social Network Members Partitioning module and Link Analysis module. Data set Pre-processing module prepares and cleans data input from social activities like email. Text pre-processing module involve processing of the enormous amount of information stored in unstructured texts cannot simply used for further. The computer handles the text as simple sequences of character strings. Therefore, specific pre-processing methods and algorithms are required for extracting useful patterns. Text mining refers to the process of extracting interesting information and knowledge from unstructured text. Text mining can be used in many areas such as information retrieval, machine learning, statistics, computational linguistics and especially data mining. The TF/IDF count the number of times each term occurs in each document and sum them all together.

The next step is clustering. Here an innovative clustering technique that purports to combine the advantages of affinity-based clustering [2] and model-based clustering. It is Similar to k-medoid clustering in that representative data points called exemplars used as centers to clusters. It is more efficient than k-medoid in the sense that the exemplars are not chosen randomly and the initial choice is close to a good solution. Here all data points are simultaneously considers as potential exemplars.

Each data point is considered as a node in a network. It recursively transmits real-valued messages along edges of the network until a good set of exemplars and corresponding clusters emerges. The magnitude of each message at any point in time reflects the current affinity that one data point has for choosing another data point as its exemplar. So we call this method as affinity propagation. It takes as input real valued similarities between data points. Similarity s(i,k) shows how well the data point with index k is suited to be the exemplar for data point i. Negative Euclidean distance used to measure similarity. One of the advantage of this method is number of original clusters do not have to be specified. It also takes an input a real number s(k,k) for each data point k so that data points with larger values of s(k,k) are more likely to be chosen as exemplars. This value is also referred to as preferences.

Two kinds of messages exchanged between data points. One is responsibility r(i,k) is sent from data point i to candidate exemplar point k. It indicates that how strongly each data point favors the candidate exemplar over other candidate exemplars. Second is availability a(i,k) is sent from candidate exemplar point k to data point i. It indicates to what degree each candidate exemplar is available as a cluster center for the data point.

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i\text{ s.t. } i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\}$$

The availability a(i,k) is set to the self-responsibility(k,k) plus the sum of the positive responsibilities candidate exemplar k receives from other points. Only the positive portions of incoming responsibilities are added, because it is only necessary for a good exemplar to explain some data points well, regardless of how poorly it explains other data points .

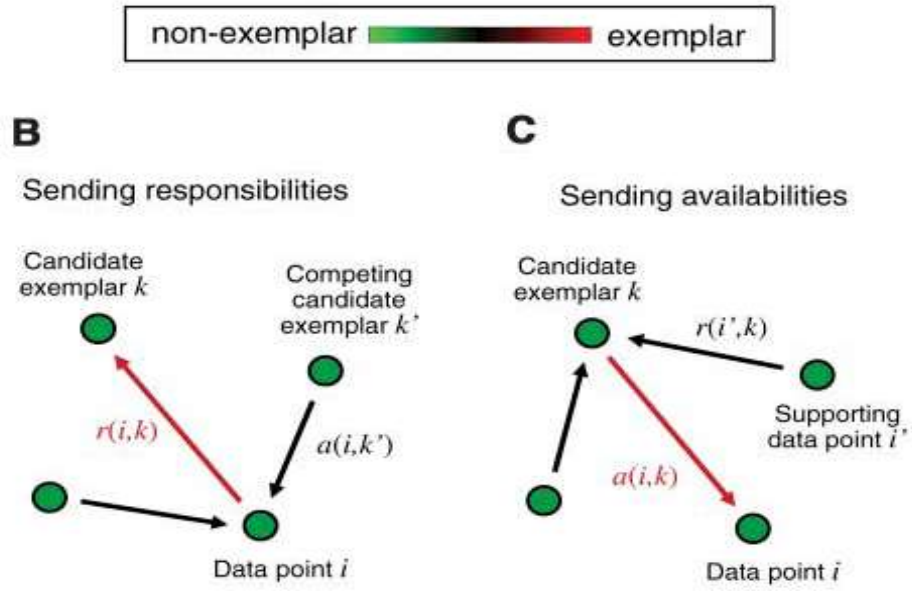

Fig.5: Affinity Propagation Is Illustrated For Two-Dimensional Data Points

Finally achieves communities by examining communication structure between users and communication between each two member user.

III. CONCLUSIONS

The study of networked communities is the expanding field of social network. The present survey has provided a state-of-the-art on existing methods. Graphs analysis will help to find a community in a larger network. Semantic community method successfully detects the communities of individuals and it provides semantic topic descriptions of these communities.

Community detection based on the action of uses will help to predicting the users actions in marketing and recommender systems. Most efficient method of detecting community is by using clustering algorithm. Affinity propagation method can efficiently determine the community than any other clustering methods. These communities have an important role in finding problems solution, managing organization and determining degree of success for people.

REFERENCES

[1] M. Coscia, F. Giannott and D. Pedreschi, "A Classification for Community Discovery Methods in Complex Networks," *Published online in Wiley Online Library* ,2011J.

[2] B. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007

[3] DingZhou1 and ,ErenManavoglu "Probabilistic Models for Discovering E-Communities", May 23-26, 2006, Edinburgh, Scotland ACM 1-59593-323-9/06/0005.

[4] Aaron Clauset, and 1 M. E. J. Newman, "Finding community structure in very large networks",National Science Foundation under grant PHY-0200909

[5] Seyed Ahmad Moosavi and Mehrdad Jalali ,"Community Detection in Online Social Networks Using Actions of Users", 978-1-4799-3351-8/14/$31.00 ©2014 IEEE

[6] Fakhri Hasanzadeh and Mehrdad Jalali, "Detecting Communities in Social Networks by Techniques of Clustering and Analysis of Communications",978-1-4799-3351-8/14/$31.00 ©2014 IEEE

[7] D. Palsetia, M. Patwary, K. Zhang, K. Lee, C. Moran, Y. Xie, D. Honbo, A. Agrawal, W. Liao, and A. Choudhary, "User-interest based community extraction in social networks," in *Proc. 6th Int. Workshop Social Network Mining and Analysis, SNAKDD*, 2012.

[8] Aron Culotta, et al., "Extracting social networks and contact information from email and the Web", In First Conference on Email and Anti-Spam, Mountain View, CA, USA. July 2005.

[9] [9] Mark Newman," Fast algorithm for detecting community structure in Networks", Phys. Rev., E, 2004.

[10] Mark Newman, "Detecting community structure in Networks", Eur. Phys. 38, 321-330, 2004.

[11] M. E. J. Newman, "Fast algorithm for detecting community structure in Networks", Phys. Rev. E 69, 066133 (2004).

[12] S. Wasserman and K. Faust," Social Network Analysis. Cambridge University Press", Cambridge (1994).