DETECTION OF VIDEO STREAMS IN CLOUD

[1]Gnana sekar v
vgs.gnanam@gmail.com
Assistant Professor

[2]Shakila M
shakisultan76@gmail.com

[3]Vaasanthi S
vaasanthi0201@gmail.com

[4]Yamini R
yaminiravi27@gmail.com

[2][3][4]UG students
Department of Computer Science and Engineering
T.J.S. Engineering College

1. Introduction

Our world is rapidly developing with many innovative technologies for enhancing the connection between the people from different corners of the world by social networks. Since, the number of users in social network is high. It is hard to find the hidden information about the users. Aim of Community detection is to identify highly connected groups of individuals or objects inside the networks, these groups are called as communities. The community detection is used to help a brand to understand the different perspective of opinion toward its product and target certain groups of people or identify influencers for the product, it can also help an e-commerce website build a recommendation system based on purchasing, and the examples are numerous.

The graph representation of real systems is community structure i.e. the Organization of vertices in clusters in which edges joining vertices of the same cluster is more and edges joining vertices of different clusters is less. Detecting communities are of great importance in all disciplines.

Earlier systems mainly concentrate about the node attribute and similarity to identify the seeds.

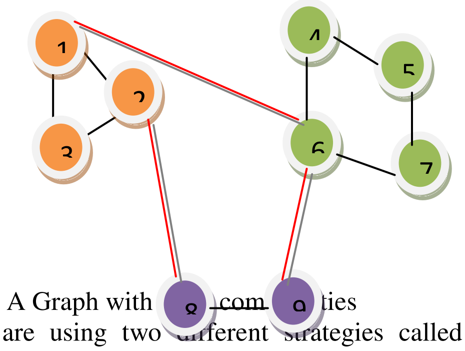

Fig 1: A Graph with two communities

We are using two different strategies called Graclus Center which calculate the centroid vertex and Spread Hub which is based on high degree vertices for finding good seeds and in seed dispersion phase we use Personalized Page Rank (PPR) algorithm to expand the seed set based on the PPR Score which is calculated for each nodes.

The Communal dilation plays an important role in the success of our algorithm by considering large sees sets than the singleton seed.

Our seeding strategies are better than the other seeding strategies and efficient to traverse through all of its neighbors and it also improves the performance and cohesiveness of the clusters, it can scale the problem of large datasets than the other existing strategies.

2. Preliminaries

2.1 Problem Statement

Given a graph G= (V,E) with a vertex set V and an edge set E, we can represent the graph as adjacency matrix A such that $A_{ij} = e_{ij}$ where e_{ij} is the edge weight. $A_{ij}=0$ if there is no edge. The goal of super imposed community detection problem is to find the super imposed clusters whose union is equal to the entire vertex set V.

2.2 Measures of cluster quality

There are some popular measures for the quality of clusters. Let us define ***links*** (C_p,C_q) to be the sum of edge weights between the vertex sets C_p and C_q.

Cut: The cut of cluster C_i is defined as the sum of edge weights between C_i and its complement V/C_i:

$$\mathit{cut}\ (C_i) = \mathit{links}\ (C_{i,V/Ci})$$

Normalized cut: It is defined by the cut with volume normalization as follows:

$$\mathbf{ncut}\ (C_i) = \frac{\mathbf{cut}\ (C_i)}{\mathbf{links}\ (C_i,V)}$$

Conductance: It is defined to be cut divided by the least number of edges incident on either set C_i or V/C_i:

$$\mathit{Cond}\ (C_i) = \frac{\mathit{cut}(C_i)}{\mathit{Min}\ (\mathit{links}\ (C_i,V),\mathit{links}\ (V/C_i,V))}$$

By definition, ***cond*** (C_i) = ***cond*** (VnC_i). The conductance of a cluster is the probability of leaving that cluster by a one-hop walk starting from the smaller set between C_i and VnC_i. Notice that cond (C_i) is always greater than or equal to ncut (C_i).

2.3 Datasets

We represent the general datasets analysis of different networks such as social networks and product networks. All the networks are connected, undirected graphs.

NAME	TYPE	NODES	EDGES
Face book	Undirected	4,039	88,234
Twitter	Directed	81,306	1,768,149
Amazon0302	Directed	2,62,111	1,234,877
Amazon0312	Directed	4,00,727	3,200,440
Amazon0505	Directed	4,10,236	3,356,834

Table 2.3 Datasets of Social and product network

In social network vertices represents users and pages, edges represents the communication path. In Amazon product network vertices represent product and edges represent co-purchasing information.

3. Literature Review

Many different approaches have been proposed for super imposed community detection. In Random Walk based label propagation algorithm [1], it uses distribution of position probability to measure the node importance. The randomness is eliminated by giving weight all labels. The performance will not be guaranteed if the distribution exceeds the predefined frequency. Tolerance granulation based community detection algorithm [2], generally all the communities are in tolerance relation i.e. There is no transitivity. The overlapped communities are merged with corresponding granule set. In this NMI accuracy is improved. In Local spectral subspaces (LOSP) [3], the local communities are identified and expanded from the given seed. With the prior information of some seeds it can detect the remaining members with high accuracy. But the strengthening of initial seed set is difficult. It also finds the multiple memberships of the individual users. The Locally Adaptive Random Transitions (LART) [4] is proposed for multiplex network which is to find communities shared by some or all layers. It uses local topological similarities between the layers. The main advantage is it can differentiate the shared and non shared communities. In Random Walk based clustering algorithm [5], the node similarity is calculated using Random Walk with Restart (RWR) which is form of Google's Page Rank algorithm. It uses both topological and knowledge based evaluation based on user defined overlapping ratio.

4. Existing System

The existing system is based on the Random Walk algorithm. In this algorithm, a graph having n number of clusters, the random walker will take more time to traverse the nodes. Because of the huge number of interconnections inside the cluster. The Random Walker start from the each node in the graph, if the random walk is short, it will stay in the same community. The random walk will return to the initial node by tracking the nodes that are closer or neighborhood to that initial node. The node is collected, which was visited by the walker is used to return to the initial node that serves as evidence that walker is belong to the same community. It is efficient but not traversed through all the nodes.

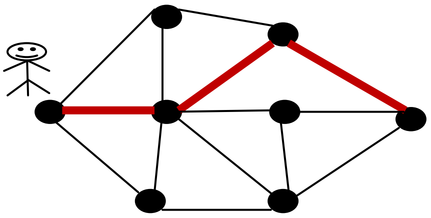

Fig 4: Random walk

5. Proposed System

The proposed system is based on the **Communal Dilated Seed Dispersion** method. In this method we are selecting good nodes from the huge number of nodes, by the following phases. First, Filtering phase involves removal of the unwanted regions which has single edge. Next is Seeding phase involves two strategies to find the good nodes. Third is Seed Dispersion phase involves PPR to further expansion of seeds. Finally Propagation phase involves expanding the seeds which was removed in the filtering phase.

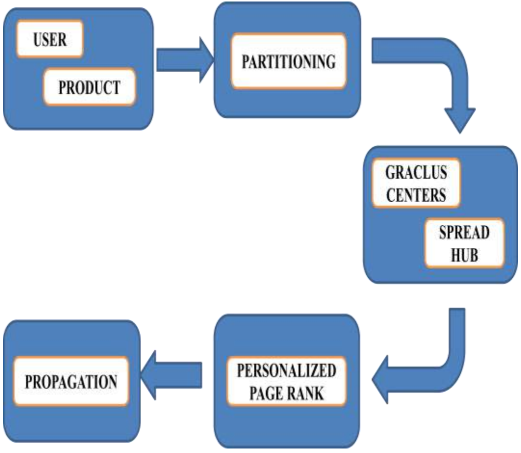

Fig 5: System Architecture

5.1. Filtering Phase

In the filtering phase, the graph partitioning method is used. In that we are removing the unwanted edges from the graph (i.e.) the graph contains the single edge. It is used to find the large connected component after removing the all single edge component. The output is called Biconnected core. The Biconnected core contains about 90% of the original graph.

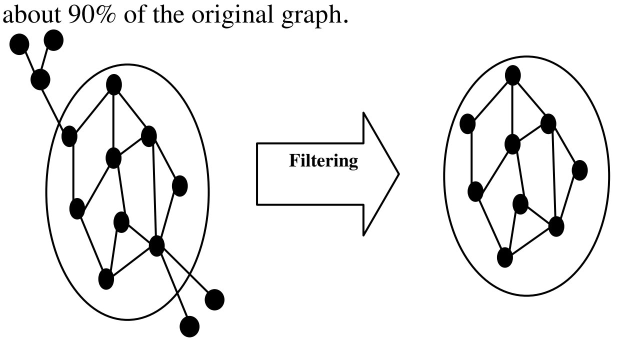

Fig 5.1: Biconnected core after filtering

5.2. Seeding Phase

This phase is used to find the good seeds with good conductance after the filtering phase. Two seeding strategies are used.

5.2.1.Graclus Centers:

From the group of clusters we are considering the central vertex among the vertices as a seed and it must be close to the cluster centroid. It is used to compute a set of seeds with small conductance. The idea here is that we need something which is close to the partition and to be good.

Algorithm 1: Sending by Graclus Centers

```
Input     : graph G, the number of seeds.
Output    : the seed set s.
Compute Non super imposed clusters $C_i$ on G.
Initialize S=$\emptyset$
for each cluster $C_i$ do
    for each vertex v€$C_i$ do
        Compute dist(v,$C_i$) using (4).
    end for
    S={argmin dist(v,$C_i$)}U S.
end for
```

5.2.2 Spread Hub

Another goal is to select a set of distributed seeds in the graph. So, they will have high coverage after we expand the sets. The Spread Hub is based on the degree of the vertices. The distance between a vertex and the cluster is inversely proportional to the degree of the vertex. Thus, the high degree vertices will have small distances to many other vertices and there should be good clusters around these vertices. So, the highest degree vertex is selected as a seed and then the neighbors are marked.

Algorithm 2 Seeding by Spread Hub

```
Input   : graph G, the number of seeds k.
Output: the seed set S.
Initialize S=$\emptyset$
All vertices in V are unmarked.
while |S|<k do
    Let T be the set of unmarked vertices with max degree.
    for each t € T do
        if t is unmarked then
            S = {t} U S.
            Mark t and its neighbors.
        end if
    end for
end while
```

5.3. Seed Dispersion Phase

The clusters have to be expanded around the seeds selected from seeding phase. A Personalized Page Rank (PPR) algorithm is proposed which is also known as Random Walk with Restart. The PPR vector with a distribution of probability α follows a step of a random walk and with probability (1-α) jumps back to the seed node. The idea is that given a set of restart nodes, we first compute the PPR vector, examine nodes in order of highest to lowest PPR score, and then return the set that achieves minimum conductance.

Algorithm 3 Seed Dispersion Phase by PPR

```
Input: graph G= {V, E}, a seed node s € S, Page Rank link-following probability parameter $0<\alpha<1$, accuracy $\epsilon>0$
Output: low conductance set C
Set T={s} U {neighbors of s}
Initialize $x_v$=0 for v € V
Initialize $r_v$=0 for v € V /T, $r_v$=1/|T| for v € T
while any $r_v$>deg (v) $\epsilon$ do
    Update $x_v= x_v + (1-\alpha) r_v$
    For each (v, u) € $\epsilon$,
        Update $r_u= r_u + \alpha\, r_v / (2 \deg(v))$
    Update $r_v = \alpha\, r_v/2$
end while
sort vertices by decreasing $x_v$/deg(v)
For each prefix set of vertices in the sorted list, compute the conductance of that set and set C to be to be the set that achieves the minimum.
```

We use entire communal of a seed node as the restart nodes rather than as a singleton nodes i.e., each seed is solely used as the restart region. We can see that the performance significantly degrades when singleton nodes are used for all seeding strategies. So, communal nodes are much better than singleton seeds on all the networks. The communal nodes also produce the low conductance communities.

The algorithm keeps two vectors of values for each vertex, i.e. x and r. The implementation uses hash tables for the vectors x and r.

5.4. Propagation Phase

In this phase, we further expand each of the communities to the regions detached in the filtering phase. Each detached whisker is connected via a bridge, we add that to all of the clusters that utilize the other vertex in the bridge.

The propagation improves the quality of the final clustering result in terms of the normalized cut metric.

Algorithm 4 Propagation Phase

```
Input: graph G, biconnected core $G_c$  communities $C_i$.
Output: Communities of G.
for each $C_i$ € C do
    detect bridges $\epsilon_{Bi}$ , attached to $C_{i.}$
    for each $b_j$ € $\epsilon_{Bi}$ do
        detect the whisker $w_j = ( v_j , \epsilon_j)$ which is attached to $b_{j.}$
        $C_i = C_i$ U $V_j$
    end for
end for
```

6. Conclusion

Our method is faster than other community detection methods. The cost is reduced by “graclus centers” for the seeds. We avoid seeding exclusively inside a dense region by using an entire communal vertex as a seed, which grows the set beyond the dense region. Thus, the communities we find likely capture a combination of communities given by the network of the original seed node. Our method is better than existing strategies in terms of run time, cohesiveness of communities and accuracy.

9. References

[1] Su Chang, Yu Yue, Xie Xianzhong and Jia Xiaotao, “A New Random-walk Based Label Propagation Community Detection Algorithm” 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.

[2] Bingjing Cai, Haiying Whui Wangang, Huiru Zheng,“An Improved Random Walk Based Clustering Algorithm for Community Detection in Complex Networks”, 978-1-4577-0653-0/11/$26.00 ©2011 IEEE.

[3] Shu Zhao, Wang Ke, Jie Chen, Feng Liu, Menghan Huang, Yanping Zhang, and Jie Tang, “Tolerance Granulation Based Community Detection Algorithm”, ISSN1007-0214 09/13 pp620-626,Volume 20,Number 26,December 2015.

[4] Kun He, Yiwei Sun, David Bindel, John Hopcroft, Yixuan Li , “Detecting Overlapping Communities from Local Spectral Subspaces”, 2015 IEEE International Conference on Data Mining.

[5] Zhana Kuncheva, Giovanni Montana,” Community Detection in Multiplex Networks using Locally Adaptive Random Walks”, 2015 IEEE/ACM.