Rank the Top-N Products in Co-Purchasing Network through Discovering Overlapping Communities Using (LC-BDL) Algorithm

Mohamed Farrag
Business Information systems Department,
Canadian International College,
Cairo, Egypt,
Mohamed_farrag @cic-cairo.com

Laila El Fangary
Information systems Department, Faculty of
Computers and Information, Helwan University,
Cairo, Egypt,
lailaelfangary@gmail.com

Mona Nasr
Information systems Department, Faculty of
Computers and Information, Helwan University,
Cairo, Egypt,
m.nasr@helwan.edu.eg

Chaimaa Salama
Information systems Department, Faculty of
Computers and Information, Helwan University,
Cairo, Egypt,
chaimaa_salama@yahoo.com

***Abstract*—online stores booming up, more and more people today are using online shopping. These networks are complex and involving massive data, giving a very strong interest to a set of techniques developed for mining these graphs enable e-retailers to make better recommendation and comprehend the buying patterns. Community detection is one of the fundamental applications that provide a solution for online stores networks, disciplines where systems are often represented as graphs. It helps to understand and model the network structure, provide a mechanism for executives to assess consumer opinion using this information to improve products, customer service and perception, Also can be useful in various applications such as rating prediction, link and Top-N recommendation, trend analysis and also be the main provider engine for recommender systems. In literature, the exponentially increasing computation time of this problem make the quality of these solutions is limited and impractical. Furthermore, most of the proposed approaches are able to detect only disjoint communities. The paper first focuses on the implementation of new clique based overlapping community detection algorithm, consists of two phases. First to enumerate the maximal cliques. The second to detect the overlapping communities among the discovered maximal cliques using three different community scales based on three different depth levels to discover the largest communities and assures high nodes coverage. While the second part of the work is to rank Top-N nodes through the discovered communities in two ways. The work provides experimental results on Amazon products co-purchasing network, most popular online stores. Clustering coefficient and cluster density are used to measure the quality. The results show that our algorithm can extract meaningful communities in different scales from this network, able to rank the top-N nodes in these different scales and revealing large scale patterns present in interaction habits of customers.**

Keywords—Graph Mining; Maximal clique problem; Overlapping community detection; Top-N recommendation

I. Introduction

Online stores booming up, the market has become quite competitive. More and more people today are using the convenience of online shopping. These networks contain massive data makes the graphs representing these networks becoming very complex. Giving a very strong interest to a set of techniques developed for mining these graphs enable e-retailers to make better recommendation and comprehend the buying patterns. Detecting communities in networks is one of the fundamental applications that provide a solution for online stores network, disciplines where systems are often represented as graphs. Communities can be considered a summary of the whole network, thus making the network easy to comprehend. The discovery of these communities can increasingly being leveraged as a powerful, low-cost tool for enterprises to drive business objectives and functions such as enhanced customer interaction, provide a mechanism for executives to assess consumer opinion and use this information to improve products, customer service and perception. Obtain more knowledge about customers. Discover new products and services. Also help in advertising, direct marketing and predict to which products or services a particular customer was likely to respond to. It also help in behavior modeling and prediction, collaborative filtering. It can be useful in various applications such as rating prediction, top-N recommendation, link recommendation and trend analysis and also be the main provider engine for recommender systems. It also help detect purchasing trends in product categories and find hidden relations between different products. The paper first focuses on the implementation of new clique based approach for fast and efficient overlapping community detection consists of two phases. Phase1 enumerate the maximal cliques. While phase2, aims to detect the overlapping communities among the discovered maximal cliques in phase1 using three different community scales based on three different depth levels to detect the largest community and assures high nodes coverage for connected network. While the second part of the work concerns to rank top-N nodes through the discovered communities by two ways, first type based on ranking nodes according to overlapping frequencies among the discovered communities, while second type based on ranking nodes according to occurrence frequencies in the adjacent sub cliques that combine the discovered communities. The work provides experimental results on Amazon products co-purchasing network, one of the most popular online stores. It focuses on how customers behave or interact with Amazon products; Focus on the communities around products and reviews. First, helps to explore and discover different communities of Amazon products co-purchasing network under different views of the Amazon products reviews. Second, rank the most important products in these communities to help and support different business objectives. Results of the proposed algorithm may help to reach one or more of the next goals to support and maximize business benefits, First it may help to capture popular products or trends over period of time. Also It may provide more efficient ways for communicating with the different customer segments, clustering can help to further make an assessment about the individual such as what activities, products, and services, customers might be interested in. also a way of giving offers to customers that behave and respond in the same manner, or assessing the overall stability of a customer base. The old-school way of commercial clustering based purely on a customer's attributes such as age, gender and address, is clearly challenged when faced with modern community detection that produces information on how customers actually link together. It may also allow us to potentially represent sets of products related by certain external events for example products that became popular at the same time. The results will consider as an important mechanism for effective recommended systems, helping also in promotions strategies, cross selling and up selling strategies, where trying to promote their products and services through this large scale online communities network. This paper is organized as follows: section two discusses background and related work, section three demonstrates the proposed algorithm and section four explains the experiment while conclusion and future work are in the last section.

II. Background and Related Work

Some real life of online networks, which can be analyzed and mining to support different business objectives like online networks built on public social networks as Facebook, Twitter and LinkedIn, product co-purchasing networks like Amazon, public online communities on a company - owned Domain, collaboration networks like, business blogs, e-mail networks, telephone networks, other environments for online communities, information networks (documents, web graphs, patents), infrastructure networks (roads, planes, water pipes, power grids), biological networks (genes, proteins), also internet traffic. [1]

Some main aspects concerning the nature and the structure of on-line networks that these network modeled by a graph that are the foremost usually used abstract data structures within the field of computer science, they enable a more complicated and wide-ranging presentation of data compared to link tables and tree structures. [1, 16, 17] A network is usually presented as a graph $G(V, E)$, where V is set of n nodes and E is set of m edges. Graph G consisting of n number of nodes denoting n individuals or the participants in the network. The connection between node i and node j is represented by the edge e_{ij} of the graph. The graph are often represented by an adjacency matrix A in which A_{ij} = 1 in case there is an edge between i and j else A_{ij} = 0 [1, 3, 5]. Another aspect is Cliques, is defined as "a set of vertices in which every pair of vertices is connected by an edge".[5] Clique is a complete sub graph of G or in different words "is a maximum complete sub graph in which all nodes are adjacent to each other" , in a clique of size k, each node maintains degree≥ $k-1$. Normally use cliques as a core or a seed to seek out larger communities. [5, 6, 12] another formal definition "A clique is a fully connected sub graph a set of nodes all of which are connected to each other. " [7, 15] K-cliques are often outlined as complete graph with k vertices. [12] K-cliques are main structures in complicated networks, and a good way to seek out community structure. [7, 12] Maximal Clique "is a clique that's contained in no larger clique". [7, 12, 14] Every maximal clique is a clique, by definition, however the other doesn't hold. Therefore there are always more cliques than maximal cliques. In different words "a clique is said to be maximal if it not contained in any other clique." [7] Adjacent k-cliques, we are able to define adjacent K-cliques by two k-cliques that share $k-1$ nodes. [7, 12] K-clique community (cluster or component), outlined as "a union of all k-cliques that may be reached from each other through a series of adjacent k-cliques." or "It is the union of all k-cliques that are k-Clique-connected to a particular k-clique." [8, 12] Real world complicated systems may be represented within the form of networks. To comprehend the in-depth structure of those systems, it's necessary to review and analyze the networks. A trivial property of those networks is community structure obtained by splitting the network into many parts, within which connection between nodes are more dense than the remainder of the network. The sets of this sort of grouping are commonly referred as communities, however additionally called clusters, cohesive groups, or modules as there is no globally accepted unique definition. One among the restrictions of graph partitioning methods is that they typically need the user to specify the number of partitions, which cannot be identified before. One solution proposed to this problem is to use goodness metric as modularity to evaluate the partition of the graph at every step. However, this is often computationally expensive and might be infeasible for massive graphs. however just in case of community detection, it's not known that how many communities are present in the network and it is not at all obligatory for them to be of same size. The community detection approach assumes that almost all of real world networks, divide naturally into groups of nodes (community) with dense connections internally and sparser connections between groups, and therefore the experimenter's job is only to find these already formed groups. The number of partitions and size of them are settled by the network itself and not set by the experimenter. So community detection is "the technique which aims to discover natural divisions of networks into groups based on strength of connection between vertices." [1, 13] No formal definition of community is universally accepted, communities will have numerous properties, and community detection has been approached from many alternative views. Community detection is one among the foremost wide researched issues. Straightforward definition "A community is a densely connected group of nodes that is sparsely connected to the rest of the network" [18], Generally spoken community as "a module or cluster is typically thought of as a group of nodes with more and/or better interactions amongst its members than between its members and the remainder of the network". [16] Primarily, community may be divided into two types; disjoint communities and overlapping communities. In disjoint communities nodes can be part of only a single community. A non-overlapping community structure or disjoint community structure may be outlined as "set of communities such that all vertices are included in exactly one community." [2] However in overlapping communities partitions aren't essentially disjoint. There might be nodes that belong to more than one community [4, 18]. Usually in any on-line network a node may be part of more than one different group or community, thus for on-line networks, overlapping community detection technique ought to be thought of disjoint community detection technique.

There have also been modifications and revisions to many methods and algorithms already proposed. Two perspectives to divide the prior work in literature, first one depending on the nature of the relation between cluster members. This perspective divided into four categories. Fist category is node centric community detection where nodes satisfy different properties as complete mutuality which means cliques, another property is reachability of members as k-clique [13]. Second category is group centric community detection (Density-Based Groups), it requires the whole group to satisfy a certain condition for example the group density greater than or equal a given threshold and remove nodes with degree less than the average degree. Third category is network centric community detection, needs to consider the connections within a network globally. Its goal to partition nodes of a network into disjoint sets, five different approaches used in network-centric community detection. First approach is clustering based on vertex similarity using Jaccard similarity and Cosine similarity. Second approach is Latent space models (multi-dimensional scaling) based on k-means clustering. Third approach is block model approximation based on exchangeable graph models. Fourth approach is spectral clustering are using minimum cut problem that the number of edges between the two sets is minimized. The fifth approach is modularity maximization by measures the strength of a community partition by considering the degree distribution. While the fourth category is hierarchy centric community detection, aims to build a hierarchical structure of communities based on network topology to allow the analysis of a network at different resolutions two representative approaches first divisive hierarchical clustering (top-down) and agglomerative hierarchical clustering (bottom-up). [3, 13] The strength of a tie can be measured by edge betweenness which is the number of shortest paths that pass along with the edge. From the second perspective we can divide the methods for detecting overlapping communities in two categories. Clique based methods and non-clique based methods. [1] In clique based methods for overlapping community detection a community can be interpreted as a union of smaller, complete (fully connected) sub graphs that share nodes. A "k-clique community can be defined as a union of all k-cliques that can be reached from each other through a series of adjacent k-cliques." [1]

One of the most widely used techniques to find overlapping communities is the clique percolation method (CPM). [9, 18] CPM is an effective algorithm in identifying overlapping module structures; it has a wide range of application in social networks and biological networks. The underlying idea of this method is the concept of a k-clique community which was defined as the union of all k-cliques (complete sub graphs of size k) that can be reached from each other through a series of adjacent k-cliques (where adjacency means sharing $k-1$ vertices). The k-clique community can be considered as a usual module (community, cluster or complex) because of its dense internal links and sparse external linkage with other part of the whole network. Then construct the overlap matrix of these k-cliques. Finally, a number of k-clique communities are discovered by analysis the overlap matrix. [9, 10] CPM algorithm first extracts all complete sub graphs of the network that are not parts of larger complete sub graphs. These maximal complete sub graphs are simply called cliques, Once the cliques are located, the clique-clique overlap matrix is prepared .In this symmetric matrix each row (and column) represents a clique and the matrix elements are equal to the number of common nodes between the corresponding two cliques, and the diagonal entries are equal to the size of the clique.

The k-clique-communities for a given value of k are equivalent to such connected clique components in which the neighboring cliques are linked to each other by at least $k-1$ common nodes. These components can be found by erasing every off-diagonal entry. [1, 8, 12] Another techniques are employed in literature as a clique based overlapping communities like the algorithm projected by Lancichinetti et al.[19] It performs a local exploration in order to find the community for each of the nodes. During this method, the nodes could also be revisited any number of times. The target was to detect local maximal based on a fitness function. Also CFinder software system was developed supporting CPM for overlapping community detection. Then Du et al. [20] proposed Comtector to detect the overlapping communities using maximal cliques. At first, all maximal cliques within the network that form the kernels of a possible community are detect. Then, the agglomerative procedure is iteratively used to add the vertices left to their nearest kernels. The obtained clusters are adjusted by merging a combine of fractional communities to optimize the modularity of the network. EAGLE is another work using agglomerative hierarchical clustering based on maximal clique algorithm has been projected by Shen et al. [21] Firstly, maximal cliques are detected, and those smaller than a threshold are neglected. Then Subordinate maximal cliques are neglected, and the remaining give the initial communities. The similarity is found between these communities, and communities are repeatedly integrated along on the premise of this similarity. This is often used until one community remains at the end. Evans et al. [22] proposed that by partitioning the links of a network, the overlapping communities is also discovered. In another work, Evans et al. [23] used weighted line graphs. In another work, Evans [24] used clique graphs to discover the overlapping communities in real-world networks. Also Greedy clique expansion [25] at the first detect cliques in a network. These cliques act as seeds for expansion along with the greedy optimization of a fitness function. A community is discovered by expanding the selected seed and performing its greedy optimization via the fitness function proposed by Lancichinetti et al. [19] another work is Cluster-overlap Newman Girvan algorithm (CONGA) was proposed by Gregory. This algorithm was based on the split- betweenness algorithm of Girvan–Newman. CONGO optimized the proposed algorithm [26], which used a local betweenness measure, giving an improved complexity. A two-phase Peacock algorithm for overlapping community detection is proposed in Gregory [27] using disjoint community-detection methods. In the first phase, the network transformation was performed using the split betweenness concept proposed earlier by the author. Within the second phase, the remodeled network is processed by a disjoint community detection algorithm, and the discovered communities were transformed back to overlapping communities of the original network. [1]

III. LC-BDL (Largest Community Based On Depth Level) Algorithm

In this work, we propose LC-BDL (largest community based on depth level) algorithm which a new clique based algorithm for overlapping community detection, LC-BDL based on the assumption of one of the first and most popular and commonly used algorithm for overlapping community detection clique percolation method (CPM). The assumption that a community is 'union of all k-cliques (complete sub graphs of size k) that can be reached from each other through a series of adjacent k-cliques", (where adjacency means sharing $k-1$ vertices). The k-clique community can be considered as a usual module (community, cluster) because of its dense internal links and sparse external linkage with other part of the whole network. In the first part of the work the proposed algorithm produce the method NMC to enumerate maximal cliques by reducing the search vertices and pruning the nodes and edges that will not be a part of a maximal clique for each node. The second part of the work LC-BDL produces three different community scales depending on the target depth level for generating these communities. First "*Restricted community scale*" in which the depth level value equal zero it means that it detects the communities among only maximal cliques of threshold size K. Second "*Flexible community scale*" in which the depth level value is variant and flexible according to business target for detecting the communities, it means that it detects the communities among maximal cliques of threshold size K and its adjacent sub cliques of size equal maximal clique size till given depth L. This may lead to enlarge the detected communities in restricted scale by integrating these communities into larger communities and detect the hidden pattern of relation among these communities was discovered in restricted community scale. Third "*Power community scale*" in which the depth level value is the maximum to detect the largest communities could be reached by test all the maximal cliques adjacent sub cliques of size equal three since that the triangle structure or 3-clique is a basic sub-structure of any clique of size is larger than three to assure that no adjacent sub cliques of a maximal clique belongs to series of adjacent sub cliques for another maximal clique that helps to detect the largest communities in a given network without restriction to threshold size K and helps to increase nodes and cliques' coverage that may be part of existing community or generate new communities in the existing graph. In the following section, we present our LC-BDL algorithm as a clique based algorithm overlapping community detection. LC-BDL algorithm consists of two phases, in the first phase the algorithm enumerate nodes maximal cliques using NMC method, while phase two aims to discover the communities among the discovered maximal cliques in phase one according to three different community scales.

Let n be the number of vertices of the input graph $G=(V,E)$ where $V=\{v_1,v_2,\dots,v_n\}$ and $E=$

$\{(v_1,v_2),(v_1,v_3),\dots(v_i,v_z),\dots,(v_{n-1},v_n)\}$ denote the set of vertices and edges, respectively. The set of vertices adjacent to a vertex v_i , the set of its neighbors is defined as $N(v_i)=\{v_z|v_i\in E\}$ and the cardinality of $N(v_i)$ is denoted by $d(v_i)$, the maximal cliques denotes by $MC\subseteq V$. The community detection problem is typically formulated as finding a partition C = $\{v_1,v_2,\dots,v_k\}$ of G, where $\forall\ vK\in G$. C is also known as a clustering of G. We use N to denote the number of resulting communities, that is $|C|=N$.

A. Phase1: Enumerate Maximal Cliques

Method: NMC (Nodes maximal cliques) for finding the maximal clique in a graph
Input: Input: Graph $G = (V,E)$
Output: Nodes maximal cliques

```
For $i = 1{:}n$
  $A_i = \{N(v_i) \cup v_i\}$
    for each node $N_j$ in Set $A_i$
      $X_j = \{d(A_i)|d(N_{j+1}) \le d(N_j)\ \forall N_j \in A_i\}$
      $X = \cup_j X_j$
    Loop
  Let $x_j$ be an elementary element of $X$
  For $j = 1{:}|X|$
    $CARD_j = |\{N_j|d(N_j) \ge x_j\ \ \forall N_j \in A_i\}$
    If $CARD_j \ge x_j$ Then
      $B_i = \{A_i|d(N_j) \ge x_j\}$
      For each node $N_k$in Set $B_i$
        $X'_k = \{d(B_i)|d(N_{k+1}) \le d(N_k)\ \forall N_k \in B_i\}$
        $X' = \cup_k X'_k$
      Loop
      Let $x_k$ be an elementary element of $X'$
     $MINS = Min(X')$
     $CARDB_i = |B_i|$
     $CARDMINB_i = |\{x_k|x_k = MINS\ \ \forall x_k \in X'\}|$
      If $CARDB_i = CARDMINB_i$ then
        $MC_i = B_i$
     Else
        $MC_i = \{B_i|x_k \ne MINS\ \ \forall\ x_k \in X'\}$
   Else
  Loop
Loop
```

NMC method aims to optimize the process of enumerating the maximal cliques to reduce time cost and enhance performance of the algorithm. To discover nodes maximal cliques NMC method reducing the search vertices and pruning the nodes and edges that will not be a part of a maximal clique for each node in two steps, first step generate set $A_i=\{N(v_i)\cup v_i\}$ then compute $d(A_i)$ equal the cardinality of $N(v_i)$ for each member in set A_i . Then set A_i sorted descending according to $d(A_i)$. The method iterate the next procedure till $CARD_j\ge x_j$ by let $CARD_j$ equal cardinality of nodes in set $\mathrm{A_i}$ that its $d(N_j)\ge x_j$ then compare $CARD_j$ against x_j, if $CARD_j$ is greater than or equal x_j the method returns set $B_i=\{A_i|d(N_j)\ge x_j\}$ by remove all nodes that will not be a part of a maximal clique for this vertex in case else the method uses the next x_j.
In second step for each node N_k in Set B_i the method assign X'_k equal $d(B_i)$ for each member in set B_i. Then let $X'=\cup_k X'_k$ Then set B_i sorted descending according to $d(B_i)$. Then three variables are used first $MINS$ equal the minimum value in X' , $CARDB_i$ equal the cardinality of set B_i and $CARDMINB_i$ equal the cardinality of set B_i where $X'_k=MINS$. Then compare $CARDB_i$ against $CARDMINB_i$, if $CARDB_i$ is equal $CARDMINB_i$ the method returns maximal clique equal set B_i case else the method exclude all vertices where its x_k equal to the $MINS$ and returns maximal clique equal set B_i where $x_k\ne MINS$. Finally define the set $MC=\{MC_1,MC_2,\dots,MC_w\}$.

B. Phase2: Discover the communities

LC-BDL algorithm aims to discover the communities among the discovered maximal cliques in phase one according to three different community scales in two steps. Step1 aims to create test cliques list and generate adjacent list among them. This may be generated according to one of three different community scales depending on the target depth level for generating these communities. First "*Restricted community scale*" in which the depth level value equal zero it means that it detects the communities among only maximal cliques of threshold size K. Second "*Flexible community scale*" in which the depth level value is variant and flexible according to business target for detecting the communities, it means that it detects the communities among maximal cliques of threshold size K and its adjacent sub cliques of size equal maximal clique size till given depth L. Third "*Power community scale*" in which the depth level value is the maximum value to detect the largest communities could be reached by testing all the maximal cliques adjacent sub cliques of size equal three since that the triangle structure or 3-clique is a basic sub-structure of any clique whose size is greater than three to assure that no sub cliques of a maximal clique belongs to series of adjacent sub cliques for another maximal clique. This helps to detect the largest communities in a given network without restriction to threshold value K. While step two aims to generate the communities among adjacent sub cliques of step one by detecting the communities' seeds then generate communities among the discovered seeds.

1) "Restricted community Scale" - Zero Depth level

Depth level $L=0$ in the restricted community scale, it means that it detects the communities among only maximal cliques of threshold size K.

a) Step 1: Generate adjacent list

The algorithm in the restricted community scale aims first to generate the test maximal cliques among only the maximal cliques was detected in first phase and according to threshold of size K where $K=3,4,5\dots t<\infty$ and depth level L where $k-L\ge 3\ \forall\ L=0,1,2,..r<\infty$. We denote test maximal cliques by TMC is the set of maximal cliques of size greater than or equal K.

$$TMC=\{MC_i|\ |MC_i|\ge K\} \qquad (1)$$

Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$ is all possible sub cliques of size equal $TMC_i - L$, where TMC_i is the i^{th} tuple in TMC, using the power set $P(.)$ we can write

$$TMCS_i = \{D_i \in P(TMC_i) | |D_i| = |TMC_i| - L\} \qquad (2)$$

Therefore the depth level L equal zero in the restricted community scale thus $TMCS_i = TMC_i$, Now define a set

$$TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\} \qquad (3)$$

Consider the set

$$BT = \{(TMCS_1, TMCS_1), (TMCS_2, TMCS_2), \dots, (TMCS_h, TMCS_h)\} \qquad (4)$$

Then LC-BDL algorithm define adjacent test list by the set AT as a list of testing each sub clique of $TMCS$ with the rest of $TMCS$ union sub clique with itself.

$$AT = \{AT \in P(TMCS) | |AT_i| = 2) \bigcup BT\}$$
$$= \{(AT_{g1}, AT_{u1}), (AT_{g2}, AT_{u2}), \dots, (AT_{gq}, AT_{uq})\} \quad (5)$$

Finally to generate the adjacent list the algorithm perform the needed computation for each tuple in AT by detecting the cardinality for each sub clique and calculate the adjacently vertices between these two sub cliques for the i^{th} tuple AT_i to be considered adjacent if one of two sub cliques share greater than or equal its cardinality -1.

b) Step2: Detect the communities

Step2 aims to generate the communities among the adjacent tuples was detected in previous step. First by detect the communities' seeds and then uses these seeds to generate the final communities.

1. Detect communities' seeds

LC-BDL algorithm creates communities' seeds by selecting the sub cliques without duplicates among the adjacent list was created in the previous step. Then retrieve back the maximal cliques instead of sub cliques.

2. Generate communities

Merge each two communities' seeds if they share one adjacent sub clique and loop until no possible merge to produce the largest community among these communities' seeds.

2) "Flexible community Scale" – variant depth level

The depth level value is variant and flexible according to business target in "*Flexible community scale*". Detecting communities in this scale based on detects the communities among only maximal cliques of threshold K and its adjacent sub cliques of size equal maximal clique size till given depth L. If the maximal clique size is seven and the depth level is 2 it means that the list will contain all adjacent sub cliques for this maximal clique of size equal five. This give the algorithm the ability to check if there is any adjacent sub clique of size $K - N$ which already are adjacent cliques because it belongs to a sequences of adjacent sub cliques from one maximal clique. To check if these sub cliques are adjacent with the rest of all other maximal cliques or its adjacent sub cliques. This gives LC-BDL algorithm the ability to detect this type of communities that consists of a series of adjacent sub cliques till given depth level.

a) Step 1: Generate adjacent list

The algorithm in the flexible community scale aims first to generate the test maximal cliques among only the maximal cliques was detected in first phase and according to threshold of size K where $K = 3, 4,5 \dots t < \infty$ and depth level L where $k - L \geq 3 \; \forall \; L = 0,1,2, .. r < \infty$. We denote test maximal cliques by TMC is the set of maximal cliques of size greater than or equal K.

$$TMC = \{MC_i | \; |MC_i| \geq K\} \qquad (1)$$

Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$ is all possible sub cliques of size equal $TMC_i - L$, where TMC_i is the i^{th} tuple in TMC, using the power set $P(.)$ we can write

$$TMCS_i = \{D_i \in P(TMC_i) | |D_i| = |TMC_i| - L\}. \qquad (2)$$

Now define a set

$$TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\} \qquad (3)$$

Then LC-BDL algorithm define adjacent test list by the set AT as a list of testing each sub clique of $TMCS$ with the rest of $TMCS$.

$$AT = \{AT \in P(TMCS) | |AT_i| = 2\}$$
$$= \{(AT_{g1}, AT_{u1}), (AT_{g2}, AT_{u2}), \dots, (AT_{gq}, AT_{uq})\} \quad (4)$$

But to enhance the performance and computation cost the LC-BDL algorithm avoids generating a test case between two sub cliques if they belong to the same $TMCS_i$, in some cases the depth level is greater than one the algorithm uses only the minimum size for the adjacent sub cliques because the goal is to check if any adjacent sub clique of a maximal clique is adjacent to another maximal clique or even belongs to a series of adjacent sub cliques to another maximal clique. So the small adjacent sub cliques is better to use instead of using all possible sub cliques for a maximal clique to enhance the performance and reduce computation cost to be available to use in real world networks.

Finally to generate the adjacent list the algorithm perform the needed computation for each tuple in AT by detecting the cardinality for each sub clique and calculate the adjacently vertices between these two sub cliques for the i^{th} tuple AT_i to be considered adjacent if one of two sub cliques share greater than or equal its cardinality -1.

b) Step2: Detecting the communities.

Step2 aims to generate the communities among the adjacent tuples was detected in previous step. First by detect the communities' seeds and then uses these seeds to generate the final communities.

1. Detect communities' seeds

LC-BDL algorithm creates communities' seeds by selecting the sub cliques without duplicates among the adjacent list was created in the previous step, then retrieve back the main maximal clique instead of adjacent sub clique.

2. Generate communities

Merge each two communities' seeds if they share one adjacent sub clique and loop until no possible merge to produce the largest community among these communities' seeds.

3) "Power community scale" - Maximum depth level

The depth level value in "*power community scale"* is maximum, This scale based on detect the communities among all the maximal cliques' adjacent sub cliques of size equal three since that the triangle structure or 3-clique is a basic sub-structure of any clique whose size is larger than three to check that no sub cliques of a maximal clique belongs to series of adjacent sub cliques for another maximal clique. It helps to detect the largest communities in a given network without restriction for threshold size K as the restricted community scale where depth level $L = 0$ or even in the flexible community scale where depth level is variant.

a) Step 1: Generate adjacent list

The algorithm in the power community scale aims first to generate the test maximal cliques among the maximal cliques was detected in first phase and according to threshold of size K where $K = 3, 4,5 \dots t < \infty$ and depth level L where $k - L \geq 3 \; \forall \; L = 0,1,2, .. r < \infty$. We denote test maximal cliques by TMC is the set of maximal cliques of size greater than or equal K.

$$TMC = MC \qquad (1)$$

Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$ is all possible sub cliques of size equal 3, where TMC_i is the i^{th} tuple in TMC, using the power set $P(.)$ we can write

$$TMCSi = \{D_i \in P(TMC_i) | |D_i| = 3\}. \qquad (2)$$

Now define a set

$$TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\} \qquad (3)$$

Consider the set

$$BT = \{(TMCS_1, TMCS_1), (TMCS_2, TMCS_2), \dots, (TMCS_h, TMCS_h)\}. \qquad (4)$$

Then LC-BDL algorithm define adjacent test list by the set AT as a list of testing each sub clique of $TMCS$ with the rest of $TMCS$.

$$AT = \{AT \in P(TMCS) | |AT_i| = 2\} = \{(AT_{g1}, AT_{u1}), (AT_{g2}, AT_{u2}), \dots, (AT_{gq}, AT_{uq})\} \qquad (5)$$

Two main enhancement used in the algorithm to enhance the performance and reduce computation cost. The LC-BDL algorithm avoids generating a test case between two sub cliques if they belong to the same $TMCS_i$. The algorithm also uses the smallest adjacent sub cliques for each maximal clique instead of testing all adjacent sub clique which increase the time and computation cost and therefore make the suitable to use on real networks. Finally to generate the adjacent list the algorithm perform the needed computation for each tuple in AT by detecting the cardinality for each sub clique and calculate the adjacently vertices between these two sub cliques for the i^{th} tuple AT_i to be considered adjacent if one of two sub cliques share greater than or equal its cardinality -1.

b) Step2: Detecting the communities.

Step2 aims to generate the communities among the adjacent tuples was detected in previous step. First by detect the communities' seeds and then uses these seeds to generate the final communities.

1. Detect communities' seeds

LC-BDL algorithm creates communities' seeds by selecting the sub cliques without duplicates among the adjacent list was created in the previous step. Then retrieve back main maximal clique instead of sub cliques.

2. Generate communities

Merge each two communities' seeds if they share one adjacent sub clique and loop until no possible merge to produce the largest community among these communities' seeds.

IV. EXPERIMENT

The dataset we use is the Amazon purchasing metadata, which consists of 548,552 products (books, DVDs, music CDs, and videos) and 7,781,990 reviews on these products. The data includes the sales rank of each product and a detailed categorization. Reviews have date information and can be associated to their reviewer by unique user ID. This data was acquired by crawling the Amazon website in 2006 and includes reviews from 1995 to 2005. The dataset source is: https://snap.stanford.edu/data/amazon-meta.html. A summary for the dataset statistics are shown in Table I.

TABLE I. AMAZON PRODUCT CO-PURCHASING NETWORK METADATA STATISTICS

Point	Value
Products	548,552
Products by product group Books	393561
Products by product group DVDs	19828
Products by product group Music CDs	103144
Products by product group Videos	26132
Reviews	7,781,990

The business objectives here is to discover the overlapping community for high rating products for a specific category under different views of the Amazon products reviews over specific time slice. Then rank the top-N products in these communities. We wrote a custom parser in order to extract out the metadata into a structured format for use in graph algorithms.

- *Preparing data*

Preparing data aims to convert metadata into structured format and to assign nodes and edges according to the business objective. *First*, splitting data into time slices according to the timestamp of the reviews. *Second*, the algorithm split dataset according to the reviewers products rank into two types of rank. The high rank for reviews rating values 5, 4 and 3 and low rank for reviews rating values 2, 1 and 0. *Finally* the algorithm uses the products as nodes and edges or links between two nodes are assigned when the same reviewer ranked two products belong to same category with same rank type for specific time slice using timestamp of the reviews. For the algorithm implementation In this experiment, the algorithm uses 109316 reviews signify the high rank reviews for product category number two on time slice 2003-2004 having 961 nodes (products) and 75240 edges denote the target business objective. Samples of edges for the dataset network are shown in Table II.

TABLE II. A SAMPLE OF EDGES FOR AMAZON PRODUCT CO-PURCHASING NETWORK

Amazon products Co-Purchasing network edges		
038550926X, 385335679	899576567, 195288076	375706410, 465078362
B00000AEFG, B00000AEFF	870830279, 195288076	375706410, 140443355
B000050I1P, B00005A8IM	B00004YWTH, B000084TTD	465078362, 140443355
195061675, 1903436036	674437764, 393092097	802428665, B0000931OL
553298046, 553568728	674437764, 375706410	B000009QPI, B000009QPE
486404277, 1551113082	674437764, 465078362	B00005T30H, B00005T30G
486404277, 192833820	674437764, 140443355	312252099, 395884179
1551113082, 192833820	393092097, 375706410	312252099, 393057658
809126591, 300083289	393092097, 465078362	312252099, 553295691
899576567, 870830279	393092097, 140443355	312252099, 451208234

A. .Phase1: Enumerate Maximal Cliques

NMC method discovers two hundred and six maximal cliques in Amazon Product co-purchasing network. A sample of three maximal cliques for phase1 result, $MC = ($ *{60175044, 738703109, 1578632080, B00001OH7V}, {60652381, 60652918, 151004838, 312111827, 312242883, 520050541, 687045169, 687278147, 784015082, 800615387, 802806015, 080536627X, 827606567, 898156122, 1561011754, 1570750343}, {60930969, 312286252, 312983220, 345447980, 375726403, 440505062, 449211479}, {61007153, 140189440, 671024078}…).*While Table III shows maximal cliques size frequencies distribution.

TABLE III. |MC| : CARDINALITY OF MAXIMAL CLIQUE, F: FREQUENCY REPRESENTS OCCURRENCE TIMES

$\lvert MC \rvert$	f	$\lvert MC \rvert$	f
3	94	10	1
4	48	11	1
5	23	12	3
6	17	14	1
7	6	16	1
8	6	18	1
9	4		

B. . Phase2: Discover the communities

1) "Restricted community Scale" - Zero depth level

a) Step 1: Generate adjacent list

The algorithm uses $K = 5$ and depth level $L = 0$ in restricted community scale and according to $TMC = \{MC_i \mid |MC_i| \geq K\}$, TMC is all maximal cliques its size greater than or equal 5 equal 64 maximal cliques among the 206 maximal cliques was discovered in previous phase for Amazon product co-purchasing network a sample for $TMC =$ *{ {60652381, 60652918, 151004838, 312111827, 312242883, 520050541, 687045169, 687278147, 784015082, 800615387, 802806015, 080536627X, 827606567, 898156122, 1561011754, 1570750343}, {006092554X, 345444388, 374199698, 765345048, 1564782131, 156947057X},...}*
Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$, where $TMCS_i = \{D_i \in P(TMC_i) \mid |D_i| = |TMC_i| - L\}$. Therefore the depth level $L = 0$ thus $TMCS_i$ = TMC_i = 64 maximal cliques Then LC-BDL algorithm define adjacent test list by the set AT and produce only 76 adjacent sub cliques among the 2080 tuples in AT as final adjacent list for *restricted community scale* using the Amazon products co-purchasing network as shown in Table IV.

TABLE IV. A SAMPLE OF STEP1 RESULT FOR RESTRICTED COMMUNITY SCALE USING THE AMAZON PRODUCT CO-PURCHASING NETWORK WHERE THE CARDINALITY OF THE FIRST SUB CLIQUE IS $|AT_{g1}|$, THE CARDINALITY OF THE SECOND SUB CLIQUE IS $|AT_{u1}|$, $d(AT_{g1}, AT_{u1})$ IS THE ADJACENT VERTICES BETWEEN THE TWO SUB CLIQUES.

AT_{g1}	$\lvert AT_{g1} \rvert$	AT_{u1}	$\lvert AT_{u1} \rvert$	$d(AT_{g1}, AT_{u1})$
1	16	1	16	16
2	6	2	6	6
3	7	3	7	7
4	5	4	5	5
5	6	5	6	6
6	6	6	6	6
7	9	7	9	9
7	9	23	10	9
8	5	8	5	5
9	18	9	18	18
9	18	45	6	5
10	8	10	8	8

b) Step2: Detecting the communities.

1. *Detect communities' seeds*

LC-BDL algorithm detects 64 seeds in adjacent tuples in AT set *{1, 2, 3,…, 64}.*

2. *Generate communities*

LC-BDL results for *restricted community scale* with depth level $L = 0$ and threshold $K = 5$ is $|C| = 54$, a sample for the first two communities C_1 = *{60652381, 60652918, 151004838, 312111827, 312242883, 520050541, 687045169, 687278147, 784015082, 800615387, 802806015, 080536627X, 827606567, 898156122, 1561011754, 1570750343},* $C_2 =$ *{006092554X, 345444388, 374199698, 765345048, 1564782131, 156947057X}.* Table V shows the discovered community cardinality.

TABLE V. A SUMMARY FOR THE DISCOVERED COMMUNITIES IN RESTRICTED COMMUNITY SCALE WHERE C_i : COMMUNITY NUMBER, $|C_i|$: COMMUNITY CARDINALITY.

C_i	$\|C_i\|$	C_i	$\|C_i\|$	C_i	$\|C_i\|$	C_i	$\|C_i\|$
1	16	15	5	29	5	43	6
2	6	16	5	30	6	44	5
3	7	17	7	31	6	45	5
4	5	18	5	32	5	46	6
5	6	19	5	33	12	47	5
6	6	20	5	34	14	48	8
7	10	21	12	35	5	49	9
8	5	22	7	36	9	50	5
9	19	23	12	37	5	51	8
10	8	24	8	38	7	52	6
11	5	25	6	39	5	53	5
12	6	26	9	40	8	54	5
13	6	27	5	41	5		
14	6	28	6	42	8		

2) "Flexible community Scale" – variant depth level

a) Step 1: Generate adjacent list

The algorithm uses $K = 5$ and since that the triangle structure or 3-clique is a basic sub-structure of any clique whose size is larger than three the depth level $L = 2$. And according to $TMC = \{MC_i|\ |MC_i| \geq K\}$, only 64 maximal cliques its size greater than or equal 5. A sample of TMC = *({60652381, 60652918, 151004838, 312111827, 312242883, 520050541, 687045169, 687278147, 784015082, 800615387, 827606567, 802806015, 080536627X, 827606567, 898156122}, {006092554X, 345444388, 374199698, 765345048}, …..)*. Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$ is $TMCS_i = \{D_i \in P(TMC_i)||D_i| = |TMC_i| - L\}$, Now define a set $TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\}$ consists 2097 sub cliques for the 64 maximal cliques when $K = 5$ and $L = 2$. Then LC-BDL algorithm define adjacent test list by the set AT and produce only 14713 adjacent sub cliques among the 1316550 tuples in AT as final adjacent list for the *flexible community scale* using the Amazon product co-purchasing network.

TABLE VI. A SAMPLE OF STEP1 RESULT FOR FLEXIBLE COMMUNITY SCALE USING AMAZON PRODUCT CO-PURCHASING NETWORK WHERE THE CARDINALITY OF THE FIRST SUB CLIQUE IS $|AT_{G1}|$THE CARDINALITY OF THE SECOND SUB CLIQUE IS $|AT_{U1}|$, D(AT_{G1}, AT_{U1}) IS THE ADJACENT VERTICES BETWEEN THE TWO SUB CLIQUES.

AT_{g1}	$\|AT_{g1}\|$	AT_{u1}	$\|AT_{u1}\|$	$d(AT_{g1}, AT_{u1})$
7	7	23	8	7
7	7	30	3	2
7	7	38	3	2
7	7	45	4	3
7	7	87	8	7
7	7	89	4	3
7	7	94	3	2
7	7	98	3	2
7	7	102	3	2
7	7	151	8	6
7	7	154	4	3
7	7	173	4	3
7	7	215	8	6
7	7	217	4	3
7	7	218	4	3

b) Step2: Detecting the communities.

1. Detect communities' seeds

LC-BDL algorithm detects 64 seeds from sub cliques was created in the previous step by retrieve back the maximal cliques instead of sub cliques in adjacent tuples in AT set.

2. Generate communities

LC-BDL results for *flexible community scale* with depth level $L = 2$ and threshold $K = 5$ is $|C| = 36$, a sample for the first two communities C_1 = *{60652381, 60652918, 151004838, 312111827, 312242883, 520050541, 687045169, 687278147, 784015082, 800615387, 802806015, 080536627X, 827606567, 898156122, 1561011754, 1570750343} and* $C_2 =$ *{006092554X, 345444388, 374199698, 765345048, 1564782131, 156947057X}*. Table VII shows the discovered community cardinality.

TABLE VII. A SUMMARY FOR THE DISCOVERED COMMUNITIES IN FLEXIBLE COMMUNITY SCALE WHERE C_i : COMMUNITY NUMBER, $|C_i|$: COMMUNITY CARDINALITY.

C_i	$\|C_i\|$	C_i	$\|C_i\|$	C_i	$\|C_i\|$
7	81	11	5	31	6
60	6	12	6	36	14
18	8	13	6	39	9
44	8	14	6	42	7
1	16	15	5	43	5
2	6	17	7	51	6
3	7	19	5	52	5
4	5	20	5	55	5
5	6	21	12	56	8
6	6	27	9	57	9
8	5	28	5	58	5
10	8	29	6	59	8

3) "Power community scale" - Maximum depth level

a) Step 1: Generate adjacent list

According to $TMC = MC$, the algorithm select the 206 maximal cliques was detected in previous phase , a sample for TMC $=$ (*{60175044, 738703109, 1578632080, B00001OH7V}, {60652381, 60652918, 151004838, 312111827, 312242883, 520050541, 687045169, 687278147, 784015082, 800615387, 802806015, 080536627X, 827606567, 898156122, 1561011754, 1570750343}, {60930969, 312286252, 312983220, 345447980, 375726403, 440505062, 449211479}, {61007153, 140189440, 671024078}…)*. Then LC-BDL algorithm generates adjacent test list by first generates $TMCS_i$ where $TMCSi = \{D_i \in P(TMC_i)||D_i| = 3\}$. Now define a set $TMCS = \{TMCS_1, TMCS_2, \dots, TMCS_h\}$. Therefore the depth level L equal max in the *power community scale* thus $TMCS$ consists of 4423 sub cliques for the 206 maximal cliques.

TABLE VIII. A SAMPLE OF 4423 TEST SUB CLIQUES FOR THE 206 MAXIMAL CLIQUES FOR POWER COMMUNITY SCALE

$TMCS$
{60175044, 738703109, 1578632080}
{60175044, 738703109, B00001OH7V}

{60175044, 1578632080, B00001OH7V}
{738703109, 1578632080, B00001OH7V}
{60652381, 60652918, 151004838}

Then LC-BDL algorithm define adjacent test list by the set AT and produces 19524 tuples cliques among the 9099900 tuples in AT as final adjacent list for the *power community scale* using Amazon product co-purchasing network.

Table IX. A sample of step1 result for adjacent tuples in power community scale using the Amazon product co-purchasing network where the cardinality of the first sub clique is $|AT_{g1}|$the cardinality of the second sub clique is $|AT_{u1}|$, $d(AT_{g1},AT_{u1})$ is the adjacent vertices between the two sub cliques.

AT_{g1}	$\lvert AT_{g1}\rvert$	AT_{u1}	$\lvert AT_{u1}\rvert$	$d(AT_{g1},AT_{u1})$
2	3	222	3	2
2	3	154	3	2
2	3	21	3	2
11	3	1720	3	2
11	3	2103	3	2
11	3	2091	3	2
11	3	2079	3	2
11	3	2430	3	2
11	3	2422	3	2
11	3	2414	3	2
11	3	1368	3	2
11	3	1738	3	2
11	3	1702	3	2
11	3	629	3	2
11	3	47	3	3

b) Step2: Detecting the communities.

3. Detect communities' seeds

LC-BDL algorithm detects 206 seeds in adjacent tuples.

4. Generate communities

LC-BDL results for power community scale is $|C| = 139$, a sample for the first three communities C_1= *{60175044, 738703109, 1578632080, B00001OH7V}*, C_2= *{60652381, 60652918, 151004838, 201021188, 312111827, 312242883, 520050541, 687045169, 687278147, 784015082, 800615387, 802806015, 804461252, 080536627X, 827606567, 898156122, 1561011754, 1570750343, 1885767870}*, C_3 = *{006092554X, 345444388, 374199698, 765345048, 1564782131, 156947057X}*. Table X shows the discovered community cardinality.

Table X. A summary for the discovered communities in power community scale where C_i : Community number, $|C_i|$: community cardinality.

C_i	$\lvert C_i\rvert$	C_i	$\lvert C_i\rvert$	C_i	$\lvert C_i\rvert$	C_i	$\lvert C_i\rvert$	C_i	$\lvert C_i\rvert$
11	108	32	5	142	4	86	3	147	3
2	19	54	5	150	4	89	3	148	3
166	19	87	5	159	4	93	3	149	3
67	14	132	5	170	4	94	3	151	3
75	14	155	5	174	4	98	3	152	3
35	13	167	5	184	4	99	3	157	3
53	9	1	4	185	4	100	3	158	3
181	9	14	4	188	4	102	3	161	3
16	8	30	4	195	4	107	3	162	3
28	8	36	4	5	3	108	3	163	3
160	8	37	4	6	3	109	3	164	3
4	7	41	4	8	3	110	3	165	3
18	7	42	4	12	3	114	3	169	3
27	7	52	4	19	3	115	3	171	3
85	7	69	4	22	3	116	3	172	3
3	6	70	4	24	3	120	3	175	3
9	6	77	4	31	3	121	3	176	3
10	6	88	4	33	3	124	3	177	3
20	6	90	4	34	3	126	3	178	3
23	6	91	4	39	3	129	3	189	3
56	6	95	4	43	3	133	3	194	3
60	6	104	4	44	3	134	3	196	3
125	6	111	4	50	3	138	3	197	3
7	5	113	4	58	3	139	3	199	3
13	5	122	4	61	3	140	3	202	3
17	5	128	4	66	3	141	3	204	3
25	5	130	4	74	3	143	3	206	3
29	5	136	4	83	3	145	3		

V. Results and Evaluation

Communities are detected for Amazon product co-purchasing network to high rank reviews for product category number two on time slice 2003-2004 using the proposed algorithm, in literature it is found that generally value of k ranges from 3 to 6. Here k value is taken as 5. NMC method in first part of the work success to discover and enumerate maximal cliques producing 206 maximal cliques. A summary for maximal cliques cardinality distribution are shown in Table III and Fig. 1.

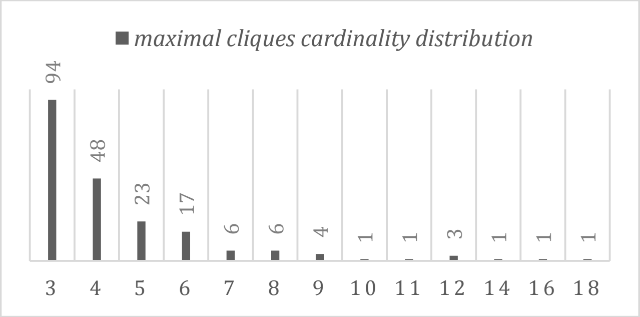


Fig. 1. A summary for maximal cliques cardinality distribution

Second part of the proposed algorithm success to produce three different scales with three different depth levels for the discovered communities. First "*restricted community scale*" in which the depth level L value equal zero it means that it detects the communities among only maximal cliques of threshold size K and depth level L equal zero. Second "*flexible community scale*" in which the depth level L value is variant and flexible according to business target for detecting the communities, it means that it detects the communities among maximal cliques of threshold size K and its adjacent sub cliques of size equal maximal clique size till given depth L. This leads to enlarge the detected communities in restricted community scale by integrating these communities into larger communities and detect the hidden pattern of relation among these communities was discovered in restricted community scale. Third "*power* *community scale*" in which the depth level value is maximum to detect the largest communities could be reached by testing all the maximal cliques adjacent sub cliques of size equal three since that the triangle structure or 3-clique is a basic sub-structure of any clique of size is larger than three. This checks that no adjacent sub cliques for a maximal clique belongs to series of adjacent sub cliques for another maximal clique. This helps to detect the largest communities in a given network without restriction to threshold size K and help to avoid neglected nodes or cliques that may be part of existing community or generate new communities in the existing graph. Initially with $k = 5$. *Restricted community scale* produce 54 communities $|\mathrm{C}| = 4$, 42 vertices are overlapped between these communities. Total 668 vertices among 961 vertices are not included in any community and the covered nodes percentage equal 30.4%. *Flexible community scale* successes to integrate the 54 discovered communities in the restricted community scale with threshold $k = 5$ and depth level $L = 2$ in 36 large communities, with the same covered nodes percentage equal 30.4%, only 27 overlapping nodes between the discovered communities. While the *power community scale* success to double the node covered ratio, make it equal 60.2% with total 579 vertices among 961 vertices are included in the discovered communities and also success to change the community structure discovered by the two previous community scales, with 84 vertices are overlapped between discovered communities. The details of community structures detected by LC-BDL algorithm for Amazon co-purchasing network for *k* value as 5 are summarized and compared in Table XI and Fig. [2-4].

TABLE XI. $D.L$: DEPTH LEVEL; K: THRESHOLD VALUE; $|AT|$: TUPLES CARDINALITY OF ADJACENT TEST LIST; $|adj(AT)|$: ADJACENT TUPLES CARDINALITY OF ADJACENT TEST LIST; S: NUMBER OF MAXIMAL CLIQUE SEEDS; $|C|$: NUMBER OF DISCOVERED COMMUNITIES; CV: NUMBER OF VERTICES COVERED; CR: % OF NODES COVERED; UVC: NUMBER OF NODES UNCOVERED; OV: NUMBER OF OVERLAPPED NODES.

Scale	$D.L$	K	$\|AT\|$	$\|adj(AT)\|$	S	$\|C\|$	CV	UVC	OV	CR
Restricted	0	5	2080	76	64	54	293	668	42	30.40%
Flexible	2	5	1316550	14713	64	36	293	668	27	30.40%
Power	Max	-	9099900	19524	206	139	579	382	84	60.20%

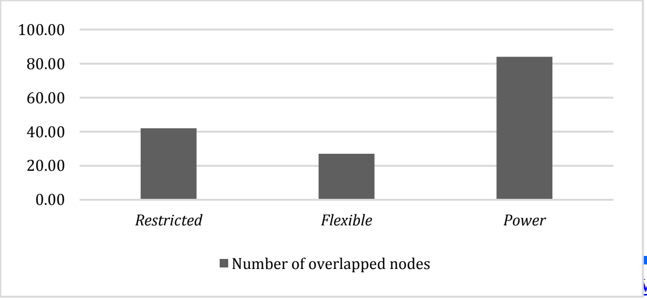


Fig.2. $|C|$ *Number of discovered communities*

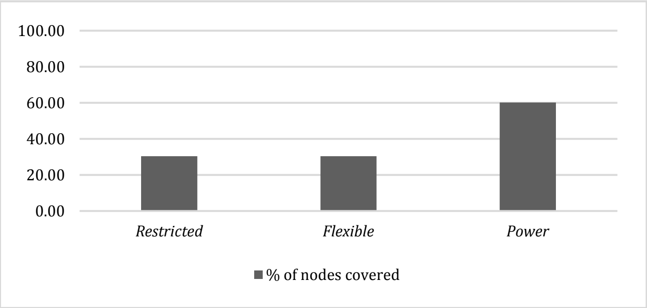
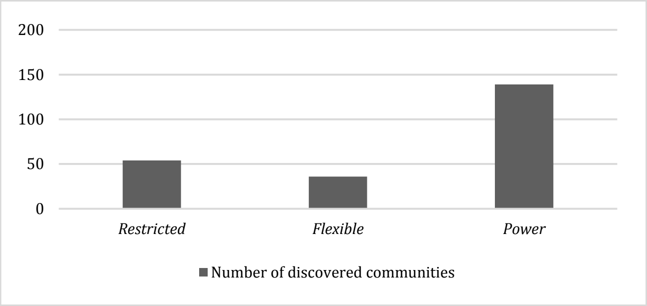


Fig.3. CR *denote % of nodes covered*

Fig.4. OV *Number of overlapped nodes*



A. Top-N nodes

The algorithm ranks the Top-N nodes through the discovered communities. First, based on the overlapping frequencies among the discovered communities while the second based on occurrence frequencies in the adjacent sub cliques that combine the discovered communities.

1) Top-N nodes based on overlapping frequencies

The number of discovered communities $|C|$=54 with total 42 overlapping frequencies among 293 covered nodes with covered ratio reach 30.40% in *restricted community scale*. The overlapping frequencies top 10 nodes are shown in Table XII.

TABLE XII. THE OVERLAPPING FREQUENCIES TOP 10 NODES IN RESTRICTED COMMUNITY SCALE.

Node	Product	*f*
0375726403	Empire Falls (Vintage Contemporaries)	13
0374199698	Middlesex: A Novel	9
038550926X	The Devil Wears Prada : A Novel	8
0385505833	Skipping Christmas	6
0060652918	Mere Christianity/The Screwtape Letters (Collector's Box Set)	5
0380817683	Rachel's Holiday	4
0446520802	The Notebook	4
0671002759	The Red Badge of Courage (Enriched Classics)	4
0312111827	Howards End (Case Studies in Contemporary Criticism)	3
0345427513	Gone for Soldiers : A Novel of the Mexican War	3

The number of discovered communities $|C|$=36 with total 27 overlapping frequencies among 293 covered nodes with covered ratio reach 30.40% in *flexible community scale*. The overlapping frequencies top 10 nodes are shown in Table XIII.

TABLE XIII. THE OVERLAPPING FREQUENCIES TOP 10 NODES IN FLEXIBLE COMMUNITY SCALE.

Node	Product	*f*
0671002759	The Red Badge of Courage (Enriched Classics)	4
0312111827	Howards End (Case Studies in Contemporary Criticism)	3
0352336862	Cooking Up a Storm (Black Lace)	3
0451528611	Anna Karenina (Signet Classics (Paperback))	3
0827606567	Jps Hebrew-English Tanakh: The Traditional Hebrew Text and the New Jps Translation	3
0060652918	Mere Christianity/The Screw tape Letters (Collector's Box Set)	2
0060925000	A Suitable Boy : Novel, A	2
0312286252	Contest	2
0312872402	Kushiel's Avatar (Kushiel's Legacy)	2
0345383451	Great Jewish Quotes	2

The number of discovered communities $|C|$=139 with total 84 overlapping frequencies among 579 covered nodes with covered ratio reach 60.20% in *power community scale*. The overlapping frequencies top 10 nodes are shown in Table XIV.

TABLE XIV. THE OVERLAPPING FREQUENCIES TOP 10 NODES IN RESTRICTED COMMUNITY SCALE.

Node	Product	*f*
0393057658	Moneyball: The Art of Winning an Unfair Game	7
0671002759	The Red Badge of Courage (Enriched Classics)	6
0345427513	Gone for Soldiers : A Novel of the Mexican War	4
0374199698	Middlesex: A Novel	4
0385335679	Kissing in Manhattan	4
0451528611	Anna Karenina (Signet Classics (Paperback))	4
B0000931OL	Think Tank	4
0060652918	Mere Christianity/The Screwtape Letters (Collector's Box Set)	3
0312111827	Howards End (Case Studies in Contemporary Criticism)	3
0312286252	Contest	3

2) *Top-N nodes based on occurrence frequencies*

The overlapping frequencies top 10 nodes are shown in Table XV.

TABLE XV. THE OVERLAPPING FREQUENCIES TOP 10 NODES IN RESTRICTED COMMUNITY SCALE.

Node	Product	*f*
0375726403	Empire Falls (Vintage Contemporaries)	17
0374199698	Middlesex: A Novel	13
038550926X	The Devil Wears Prada : A Novel	11
0385505833	Skipping Christmas	9
0380817683	Rachel's Holiday	6
0060652918	Mere Christianity/The Screwtape Letters (Collector's Box Set)	5
0352336862	Cooking Up a Storm (Black Lace)	5
0435905252	Things Fall Apart (African Writers Series)	5
0446520802	The Notebook	5
0679446230	Things Fall Apart (Everyman's Library (Cloth))	5

The overlapping frequencies top 10 nodes are shown in Table XVI.

TABLE XVI: THE OVERLAPPING FREQUENCIES TOP 10 NODES IN FLEXIBLE COMMUNITY SCALE.

Node	Product	*f*
0375726403	Empire Falls (Vintage Contemporaries)	523
0374199698	Middlesex: A Novel	457
038550926X	The Devil Wears Prada : A Novel	449
0385505833	Skipping Christmas	313
0312111827	Howards End (Case Studies in Contemporary Criticism)	291
0380817683	Rachel's Holiday	253
1558322434	Party Nuts! : 50 Recipes for Spicy, Sweet, Savory,and Simply Sensational Nuts that Will Be the Hit of Any Gathering	246
0060652918	Mere Christianity/The Screwtape Letters (Collector's Box Set)	226
0827606567	Jps Hebrew-English Tanakh: The Traditional Hebrew Text and the New Jps Translation	188
0446343455	Tourist Season	170

The overlapping frequencies top 10 nodes are shown in Table XVII.

TABLE XVII. THE OVERLAPPING FREQUENCIES TOP 10 NODES IN RESTRICTED COMMUNITY SCALE.

Node	Product	*f*
0375726403	Empire Falls (Vintage Contemporaries)	408
038550926X	The Devil Wears Prada : A Novel	365
0374199698	Middlesex: A Novel	364
0312111827	Howards End (Case Studies in Contemporary Criticism)	251
0385505833	Skipping Christmas	242
1558322434	Party Nuts! : 50 Recipes for Spicy, Sweet, Savory,and Simply Sensational Nuts that Will Be the Hit of Any Gathering	214
0380817683	Rachel's Holiday	210
0060652918	Mere Christianity/The Screwtape Letters (Collector's Box Set)	187
0827606567	Jps Hebrew-English Tanakh: The Traditional Hebrew Text and the New Jps Translation	151
0446530522	Cane River	148

The products results is too similar between the two types of rank it differ in order in few times and sometimes new products were discovered.

The LC-BDL set two types of parameters to ensure the quality of the goodness and performance metrics, while our algorithm detects cliques, adjacent k-cliques and overlapping communities, which all have clear definitions so the evaluation will depend on verify whether extracted communities satisfy the definition or not. While to evaluate the suitability and validity of our proposed algorithm in identifying the overlapping community detection in large scale networks, the average clustering coefficient and cluster density are used. LC-BDL algorithm with k value as 5, LC-BDL algorithm have very high ratio for restricted, flexible and power community scale. The details are summarized and compared in Table XVIII and Fig. [5-6].

TABLE XVIII. $|C|$ NUMBER OF DISCOVERED COMMUNITIES; $Density$: % OF THE COMMUNITY DENSITY; $Cluster\ Coefficient$: % OF THE COMMUNITY CLUSTER COEFFICIENT.

Scale	$\|C\|$		$Density$	$Cluster\ Coefficient$
Restricted	54	Minimum	0.56	0.84
		Maximum	1.00	1.00
		Average	0.97	0.98
Flexible	36	Minimum	0.14	0.81
		Maximum	1.00	1.00
		Average	0.96	0.99
Power	139	Minimum	0.09	0.80
		Maximum	1.00	1.00
		Average	0.98	0.99

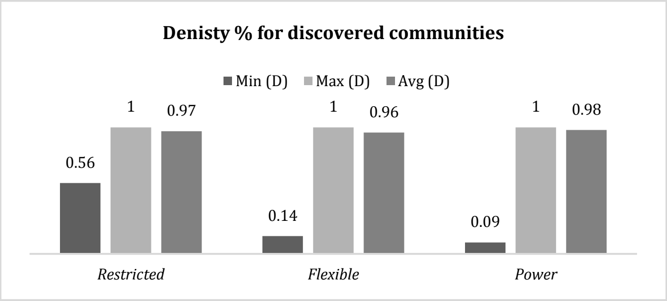


Fig.5. % of the density (D) for discovered communities.

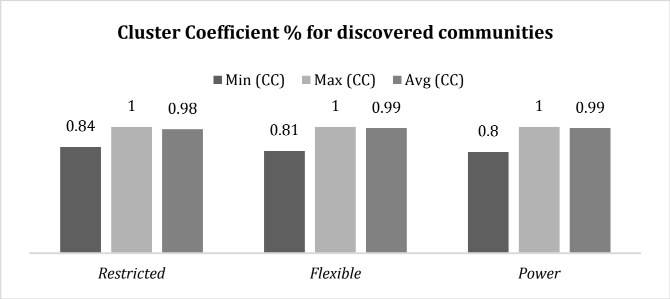


Fig.6. % of the cluster coefficient (CC) for discovered communities.

VI. CONCLUSION AND FUTURE WORK

In this work overlapping communities are identified using Amazon product co-purchasing network to high rank reviews for product category number two on time slice 2003-2004. New proposed clique based overlapping community detection algorithm has been studied. To quantify the discovered community structure the average clustering coefficient and cluster density are used. In the first part of work, a new method is proposed to enhance the process of enumerating maximal cliques by pruning specific nodes and edges. The proposed NMC method based on enumerating vertices maximal cliques by reducing the search vertices in two steps, first selects only the adjacent vertices for the tested vertex and excludes the vertices cannot be a part of existing maximal clique for this vertex. Then generate maximal clique among the rest vertices according to simple and native mathematical computation and makes the process of enumerating maximal cliques fast and efficient. Hence, for large graphs many nodes and edges will be pruned, which will reduce the computation drastically. In second part of the work, The proposed algorithm efficiently detects overlapping communities using three different community scales based on three different depth level to detect the largest community in given network and assures high vertices coverage for connected network. It is observed that based on the average clustering coefficient, cluster density and vertices coverage, proposed method gives a clear community structure. Finally, the work ranks the top-N nodes that belongs to the products that have high rank reviews for category number two on time slice 2003-2004. It can be concluded from the result that the community structure depends on the depth level for these communities and given threshold K. The community structure discovered in *restricted community scale* integrated into larger communities in *flexible community scale* and new communities discovered with high vertices cover ratio while using *power community scale*.

The proposed algorithm success to capture popular products or trends over period of time. The results will consider as an important mechanism for effective recommended systems, helping also in promotions strategies, cross selling and up selling strategies, where trying to promote their products and services through this large scale online communities network.

Overlapping community detection is still a challenge. Though there are several proposed methods, but most of them not applicable to use for real large scale graphs due to the massive data for these graphs. Taking a huge amount of processing time. So emphasis should be given to effective algorithms which will be able to detect communities in large scale online networks in allowable time. In this work only un-weighted and undirected network has been taken into consideration. In future weighted and directed networks are needed to be considered for community detection. Also not covered vertices in the network may be assigned to the discovered communities using one of similarity measure to increase the vertices cover ratio. It is a suggested to apply the proposed LC-BDL algorithm using different data domains and study its accuracy and capacity on different scopes and natures.

REFERENCES

[1] Bedi, Punam, and Chhavi Sharma. "Community detection in social networks." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 6.3 (2016): 115-135.

[2] Cuvelier, Etienne, and Marie-Aude Aufaure. "Graph mining and communities detection." Business Intelligence. Springer Berlin Heidelberg, 2012. 117-138.

[3] Adedoyin-Olowe, Mariam, Mohamed Medhat Gaber, and Frederic Stahl. "A survey of data mining

techniques for social media analysis." arXiv preprint arXiv:1312.4617 (2013).

[4] Afsarmanesh, Nazanin, and Matteo Magnani. "Finding overlapping communities in multiplex networks." arXiv preprint arXiv:1602.03746 (2016).

[5] Zafarani, Reza, Mohammad Ali Abbasi, and Huan Liu. Social media mining: an introduction. Cambridge University Press, 2014.

[6] McCreesh, Ciaran, et al. "Clique and constraint models for maximum common (connected) subgraph problems." International Conference on Principles and Practice of Constraint Programming. Springer International Publishing, 2016.

[7] Reid, Fergal, Aaron McDaid, and Neil Hurley. "Percolation computation in complex networks." Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on. IEEE, 2012.

[8] Palla, Gergely, et al. "k-clique Percolation and Clustering." Handbook of Large-Scale Random Networks. Springer Berlin Heidelberg, 2008. 369-408.

[9] Wang, Jianxin, et al. "Identifying protein complexes from interaction networks based on clique percolation and distance restriction." BMC genomics 11.2 (2010): S10.

[10] McDaid, Aaron, and Neil Hurley. "Detecting highly overlapping communities with model-based overlapping seed expansion." Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on. IEEE, 2010.

[11] Zachary, Wayne W. "An information flow model for conflict and fission in small groups." Journal of anthropological research 33.4 (1977): 452-473.

[12] Palla, Gergely, et al. "Uncovering the overlapping community structure of complex networks in nature and society." arXiv preprint physics/0506133 (2005).

[13] Nandi, G., and A. Das. "A survey on using data mining techniques for online social network analysis." Int. J. Comput. Sci. Issues (IJCSI) 10.6 (2013): 162-167.

[14] Pattabiraman, Bharath, et al. "Fast Algorithms for the Maximum Clique Problem on Massive Sparse Graphs." WAW. 2013.

[15] Palsetia, Diana, et al. "Clique guided community detection." Big Data (Big Data), 2014 IEEE International Conference on. IEEE, 2014.

[16] Leskovec, Jure, Kevin J. Lang, and Michael Mahoney. "Empirical comparison of algorithms for network community detection." Proceedings of the 19th international conference on World Wide Web. ACM, 2010.

[17] Yang, Jaewon, Julian McAuley, and Jure Leskovec. "Community detection in networks with node attributes." Data Mining (ICDM), 2013 IEEE 13th international conference on. IEEE, 2013.

[18] Harenberg, Steve, et al. "Community detection in large-scale networks: a survey and empirical evaluation." Wiley Interdisciplinary Reviews: Computational Statistics 6.6 (2014): 426-439.

[19] Lancichinetti, Andrea, Santo Fortunato, and János Kertész. "Detecting the overlapping and hierarchical community structure in complex networks." New Journal of Physics 11.3 (2009): 033015.

[20] Du, Nan, et al. "Community detection in large-scale social networks." Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. ACM, 2007.

[21] Shen, Huawei, et al. "Detect overlapping and hierarchical community structure in networks." Physica A: Statistical Mechanics and its Applications 388.8 (2009): 1706-1712.

[22] Evans, T. S., and Renaud Lambiotte. "Line graphs, link partitions, and overlapping communities." Physical Review E 80.1 (2009): 016105.

[23] Evans, Tim S., and Renaud Lambiotte. "Line graphs of weighted networks for overlapping communities." The European Physical Journal B-Condensed Matter and Complex Systems 77.2 (2010): 265-272.

[24] Evans, Tim S. "Clique graphs and overlapping communities." Journal of Statistical Mechanics: Theory and Experiment 2010.12 (2010): P12037.

[25] Lee, Conrad, et al. "Detecting highly overlapping community structure by greedy clique expansion." arXiv preprint arXiv:1002.1827 (2010).

[26] Gregory, Steve. "A fast algorithm to find overlapping communities in networks." Machine learning and knowledge discovery in databases (2008): 408-423.

[27] Gregory, Steve. "Finding overlapping communities using disjoint community detection algorithms." Complex networks (2009): 47-61.

[28] Adamcsek, Balázs, et al. "CFinder: locating cliques and overlapping modules in biological networks." Bioinformatics 22.8 (2006): 1021-1023.