

# OCMiner: A Density-Based Overlapping Community Detection Method for Social Networks

Sajid Yousuf Bhat and Muhammad Abulaish\*

Department of Computer Science, Jamia Millia Islamia (A Central University)

Jamia Nagar, New Delhi-25, India

E-mails: s.yousuf.jmi@gmail.com, mAbulaish@jmi.ac.in

## Abstract

Community detection is an important task for identifying the structure and function of complex networks. The task is challenging as communities often show overlapping and hierarchical behavior, i.e., a node can belong to multiple communities, and multiple smaller communities can be embedded within a larger community. Moreover, real-world networks often contain communities of arbitrary size and shape, along with outliers. This paper presents a novel density-based overlapping community detection method, **OCMiner**, to identify overlapping community structures in social networks. Unlike other density-based community detection methods, **OCMiner** does not require the neighborhood threshold parameter ( $\varepsilon$ ) to be set by the users. Determining an optimal value for  $\varepsilon$  is a longstanding and challenging task for density-based clustering methods. Instead, **OCMiner** automatically determines the neighborhood threshold parameter for each node locally from the underlying network. It also uses a novel distance function which utilizes the weights of the edges in weighted networks, besides being able to find communities in un-weighted networks. The efficacy of the proposed method has been established through experiments on various real-world and synthetic networks. In comparison to the existing state-of-the-art community detection methods, **OCMiner** is computationally faster, scalable to large-scale networks, and able to find significant community structures in social networks.

## 1 INTRODUCTION

An important mesoscopic structure in social networks which can often be closely related to the functional units of any system is the community. A community is defined as a group of nodes that share similar properties or connect to each other via selected relations [62]. In a community, nodes are relatively densely connected to each other, but sparsely connected to other dense groups in the network [15]. Due to increasing popularity of online social networks (OSNs) and their applications, community mining research has received a lot of attention in recent past and the field is still rapidly evolving. Numerous methods based on spectral clustering [11, 46, 55], partitional clustering [33], mathematical programming [1], and latent space clustering

---

\*Corresponding author. E-mail: abulaish@ieee.org

[18] along with modularity-based algorithms [8, 37] and likelihood-based algorithms [7] have been developed for community detection in social networks. Community detection in a network depends on various factors, including whether the definition of community relies on global or local network properties, whether nodes can simultaneously belong to several communities, whether link weights are utilized, whether outliers are considered, and whether community definition allows for hierarchical structure.

The fact that nodes in a network can belong to more than one community, and a solution based on  $k$ -clique percolation given by Palla et al. [38] have resulted in an increased attention towards the problem of overlapping community detection in social networks. Although most of the methods consider overlap of communities at boundaries, some methods allow central vertices of communities to overlap, making the characterization of overlapping vertices unclear [15]. Here, We argue that a central vertex of a community can also be a boundary vertex of another community during an overlap in a real-world network.

Besides overlapping communities, real-world social networks often show a hierarchical organization in their community structure. In such cases, multiple smaller communities at lower levels form a larger community at a higher level, or a community at lower level may be a part of even larger communities at higher levels. It thus becomes important to identify both overlapping community structures and their hierarchical organization from such networks to provide an appropriate representation of communities. Hierarchical clustering is a well-known technique used in social network analysis [52, 44] to naturally create a hierarchical tree of partitions, called dendrogram. However, such method does not consider overlaps and produces all possible partitions based on the similarity measure used, without stressing on the quality of identified community structures. Recently, a class of community detection methods [27, 41], called multi-resolution method, has started to evolve with a general property of having a tunable parameter to adjust the characteristic size of communities to be detected. Varying the value of resolution parameter enables such methods to detect community structures at varying levels of resolutions and thus form a hierarchical organization of community structures for a network.

Considering the case of OSNs like Facebook, and Twitter, community structures have mostly been analyzed using traditional community detection techniques over un-weighted social graphs representing explicit relations (friends, colleagues, etc.) of users. However, in order to identify functional communities in OSNs, it is necessary to take users interaction data (posts, blogs, chats, comments, etc.) into consideration as well. Through these interactions users gradually form social groups/communities based on shared values and interests that are quite different from traditional communities formed on the basis of geographical locations [54]. Analyses of Wilson et al. [56] and Viswanath et al. [51] on Facebook friendship and interaction data reveal that most of the users interact only with a small subset of their declared social group. This highlights that only a subset of declared social group actually represents interactive relationships. Their results demonstrate that a large part of interactions for majority of the users occur only across a small subset (as low as 20%) of their declared social group (friends). On a 100% fraction line, it has been seen that nearly all users can attribute all of their interactions to only 60% of their friends, and for majority of the users all their interactions are reciprocated. Considering interaction degrees of the nodes in OSNs, likelihood of nodes to link to other nodes of similar degree is more than the friend network. This means that nodes in interaction network show more assortativity than the friend network, and places it close to known social networks. These findings suggest that social network based systems should be based on activity network, rather than on friend network. Activity network of OSNs can be treated as a weighted graph, and a community detection

algorithm can exploit weighted links to identify communities in the network.

This paper proposes a density-based method, **OCMiner**, which is an extended and improved version of our previously published work [2], to identify overlapping community structures in online social networks. It also provides heuristics to automatically determine a good approximation for the input parameter  $\eta$ . However, the current work is also different than our other previous work [3] which mainly aims to track evolution of dynamic communities. The proposed method is along the lines of the **SCAN** [60], **DENGRAPH** [14], and other recent density-based community detection methods like **GSK** [48] and **CHRONICLE** [22] that are based on **DBSCAN** [13]. These methods find dense communities, and also detect outliers and hubs in networks. In addition to these properties, **OCMiner** has the following features.

- **OCMiner** incorporates a novel distance function, which utilizes link weights of the interaction graph (if available) of an underlying network; besides able to find communities in un-weighted networks.
- **OCMiner** does not need the neighborhood threshold  $\varepsilon$  (mostly difficult to determine for density-based community detection methods) to be specified by the users manually. Instead, it automatically determines a local version of the same for each node locally from the underlying network using a simple but effective approach. Moreover, it also provides a heuristic to find good approximation for the second parameter  $\mu$  (minimum-number-of-points) required for density-based community detection methods.
- Unlike related density-based methods, **OCMiner** finds overlapping community structures from social networks using a density-based approach, which to the best of our knowledge is the first attempt to do so.
- Tuning the only input parameter for **OCMiner** enables it to find hierarchical organization of overlapping communities at different levels of granularity. This property places it in the multi-resolution class of community detection methods.
- **OCMiner** is computationally faster and naturally scalable to large social networks.

The rest of the paper is organized as follows. Section 2 presents a review of the state-of-the-art techniques for community detection in social networks. Section 3 defines distance function and presents the procedural detail of **OCMiner**. Section 4 discusses the overlapping community detection characteristics of **OCMiner**. Section 5 provides experimental setup and evaluation results. Section 6 discusses the input parameter  $\eta$ , followed by the complexity analysis of **OCMiner** in section 7. Finally, section 8 concludes the paper with future directions of work.

## 2 RELATED WORK

Traditional approaches for community identification in networks use graph partitioning methods that divide vertices of a network into a predefined number of groups in such a way that the number of edges lying between the groups is minimal. Kernighan-Lin algorithm [21] is one of the earliest known partitioning methods. Partition-based clustering [34] is another technique extended for community detection in networks. Given a set of nodes and a predefined value,  $k$  (number of clusters to be found), the problem is to divide the nodes into  $k$  clusters that optimizes a given cost function. However, the main drawback of these methods is the requirement of the number of clusters apriori [37].

Hierarchical clustering is another well-known technique used in social network analysis [44, 52]. Starting with a partition in which each node is in its own community or all nodes are in the same community, clusters are merged or split according to a topological measure of similarity between nodes. Some of the previously proposed similarity functions in the context of social networks include *jaccard*, *cosine*, and *topological overlap*. Based on the sociological notion of betweenness centrality, Girvan and Newman [16] proposed a divisive hierarchical clustering algorithm for community detection, which calculates the betweenness of all edges in the network and removes the one with the highest betweenness value. The process continues until there is no edge remaining or a stopping criterion is met. But, the method does not provide a measure to determine the best split of communities in a network. Later on, Newman and Girvan [37] proposed modularity  $Q$  to measure the quality of a division of a network into groups or communities. The idea of modularity  $Q$  is to compare the number of links inside communities to the expected number of links in a random reference network containing no community structure. High values of  $Q$  indicate network partitions in which more of the edges fall within groups than expected by chance. Later, for many methods, modularity  $Q$  became an objective function to be maximized leading to modularity optimization-based methods for community detection. Recently Li [30] proposed a modularity based community detection method using a nonlinear programming method for modularity optimization. Although modularity optimization methods have been proved to be highly effective in practice for community evaluation in both weighted and un-weighted networks,  $Q$  measure suffers with some major problems. Firstly, modularity requires information about the entire structure of the graph, which is unrealistic in case of large networks like the World Wide Web. Secondly, modularity-based methods have a resolution limit and may fail to identify smaller (possibly important) communities. As a solution to the first problem, Clauset [6] proposed a measure for local community structure, called local modularity, for graphs which lack global knowledge. Similarly, Radicchi et al. [40] proposed a divisive hierarchical method, where links are iteratively removed based on the value of their edge clustering coefficient, which is a local measure. This approach involves less computation than that of edge betweenness used in [16] and thus yields a significant improvement in the complexity of the algorithm. Moreover, the stopping criterion of the procedure depends on the properties of the communities themselves and not on the values of a quality function like modularity. Sun et al. [50] introduce maximizing *modularity-intensity* as a solution for the resolution problem of simple modularity measure to measure the cohesiveness of a network community. Extending DBSCAN algorithm [13] to undirected and un-weighted graph structures, Xu et al. [60] proposed SCAN (Structural Clustering Algorithm for Networks) to find clusters, hubs, and outliers in large networks based on structural similarity, which uses the neighborhood of vertices as clustering criteria. CHRONICLE [22] is a two stage extension of SCAN to detect dynamic behavior of communities in a dynamic network. Similarly, considering only the weighted interaction graph of online social networks, Falkowski et al. [14] extended DBSCAN algorithm to identify community structures.

The basic idea behind density-based clustering methods based on DBSCAN is that if the neighborhood of a given radius  $\varepsilon$  of a node  $p$  (i.e., the set of nodes in a network each with a *distance* from node  $p$  less than or equal to a specified threshold  $\varepsilon$ ) contains more than  $\mu$  nodes, then a new cluster with  $p$  as a core-node is created. The process then iterates to find density-reachable nodes from this core-node and defines a density-connected cluster using density-connectivity relations between the nodes [13]. Some important features of density-based community detection methods include less computations, detection of outliers, and natural scalability to large networks. However, they also suffer with some limitations. The main drawback

of traditional density-based community detection methods is the requirement of the global neighborhood threshold  $\varepsilon$  and minimum cluster size  $\mu$  to be specified by the users. They are particularly sensitive to  $\varepsilon$  parameter, which is difficult to determine apriori. As reported in [48], automatic determination of an optimal value for  $\varepsilon$  parameter required by the density-based clustering methods (e.g., DBSCAN and SCAN) is a long-standing and challenging problem. GSK method proposed in [48], which is based on an extension of the cosine structural similarity of equation, reduces the number of possible values for  $\varepsilon$  significantly by considering only the edge weights of a Core-Connected Maximal Spanning Tree (CCMST) of the underlying network. Similarly, Huang et al. [19] proposed a two-stage parameter-free extension of density-based clustering SHRINK by first finding smaller communities using the highest local structural similarity value of  $\varepsilon$  for a pair of nodes and a constant value for  $\mu$ , and then iteratively optimizing the modularity measure [37] upon joining these smaller communities. In first stage, it uses a density-based approach to detect *micro-communities* by considering dense pairs (i.e., pairs of nodes whose structural similarity is largest among their adjacent neighbor nodes). In second stage, it iteratively joins micro-communities by considering the gain in modularity [37]. Motivated by the fact that entities in a network can simultaneously belong to multiple communities, the issue of detecting overlapping communities has received a lot of attention in recent past. The most popular method for identifying overlapping communities is the Clique Percolation Method (CPM) proposed by Palla et al. [38], which is based on the concept of  $k$ -clique, i.e., a complete subgraph of  $k$  nodes. As an enhanced variation of CPM, Kumpula et al. [24] developed a Sequential Clique Percolation (SCP) algorithm, which involves detecting  $k$ -clique communities by sequentially inserting edges of the underlying graph one by one, starting from an initial empty graph. A different method combining spectral mapping, fuzzy clustering and optimization of a quality function has been proposed in [63]. They have presented a possible embedding of vertices of an arbitrary graph into a  $d$ -dimensional space using spectral mapping to utilize fuzzy  $c$ -means algorithm on graphs for identifying overlapping communities. However, the eigenvector calculations involved in their algorithm render it computationally expensive to use on larger networks. In [53], the authors first partition a network into seed groups of overlapping community structures using existing spectral clustering method. A locally optimal expansion process is then applied to greedily optimize Newman’s modularity [37] measure. In [35], the authors presented an overlapping community detection method, MOSES, by combining local optimization with overlapping stochastic block modeling using a greedy maximization strategy. Here communities are created and deleted, and nodes are added or removed from communities, in a manner that maximizes a likelihood objective function. Sun et al. [49] present a method based on fuzzy relational model for clustering network structures into overlapping communities.

Besides overlapping communities, networks often show a hierarchical organization in their community structures, where multiple smaller communities are embedded within larger communities or a community may be a part of even larger communities. In order to provide appropriate information about the modular structure of a network, it is desirable to detect overlapping communities along with their hierarchical organization. A two-stage algorithm, EAGLE, proposed by Shen et al. [45] for detecting overlapping and hierarchical community structures in a network involves identifying all maximal cliques in the network, which along with each subordinate vertex (single vertices that do not belong to any clique) are taken as an initial set of communities. A dendrogram is then created in an iterative way using an agglomerative approach. In second phase, a proper cut-point for the dendrogram is determined by finding the maximal value of an extended modularity measure which also considers the number of communities to which a node belongs to.

Reichardt and Bornholdt [41] have shown that it is possible to reformulate the problem of community detection as a problem of finding the ground state of a spin glass model for detecting hierarchical and overlapping community structures from complex networks. The energy of the spin system is equivalent to the quality function of the clustering with the spin states being the group indices. This implies that edges should connect vertices of the same spin state, whereas vertices of different spin states should be ideally disconnected. A single parameter  $\gamma$  relates the weight given to missing and existing links in the quality function and allows for an assessment of overlapping and hierarchical community structures. In [27], the authors have proposed a method for simultaneously uncovering both hierarchical and overlapping community structures based on local optimization of a fitness function. The method performs a local exploration of the network, searching for the natural community of each node (community structure is revealed by peaks in the fitness histogram). The procedure enables each node to be included in more than one module, leading to a natural description of overlapping communities. Furthermore, the variation of a resolution parameter, determining the average size of the communities, allows exploring hierarchical levels of the community structures in a network. Along the line of the CPM [38], Kumar et al. [23] proposed a method, (HOC), to identify hierarchical and overlapping communities by finding maximal cliques in the underlying network. However, unlike CPM, HOC uses *topological overlap* criteria of equation 14 to define similarity between two arbitrary nodes in a network. For HOC, if two nodes have their neighborhood *topological overlap* (equation 14) greater than a threshold,  $\alpha$ , they belong to the same community. The community detection framework, **Infomap**, presented in [43] reformulates community detection as minimizing the description length of a random walk across the network. The total description length consists of the length for encoding community transitions and the length for encoding movements within communities. **Infomap** considers smaller description for the trajectory of random walk to be more reasonable for defining a community partition. In [42], **Infomap** is extended for networks with hierarchical community structure. In [28], the authors presented **OSLOM** which locally optimizes the statistical significance of clusters defined with respect to a random graph generated by a configuration model during community expansion. **OSLOM** is able to detect a hierarchical community structure by reapplying the algorithm on intermediate super-networks of detected communities. Recently, [59, 58] proposed an overlapping community detection method **SLPA** based on a label propagation approach. Here, labels are propagated between nodes according to pairwise interaction rules. Each node is associated with a memory which stores the received label(s). The probability of observing a label in a node's memory is interpreted as the membership strength. **SLPA** also considers the directed and weighted nature of networks to find overlapping community structures. The methods proposed in [27], [23], [41], and [59, 58] can be considered as instances of the class multi-resolution methods that generally have a freely tunable parameter (resolution parameter) which allows to set the characteristic size of the clusters to be detected. This enables them to extract communities at varying levels of resolutions and thus form a community hierarchy.

Another major challenge related to the area of community analysis in social networks is tracking the evolution of communities in dynamic social networks. The evolution of dynamic networks is mainly driven by the addition of new nodes and links to the network. The mapping of the communities across a time-step is traditionally performed by checking if any two communities across a time-step (identified individually for each time-step separately) share any core-nodes. However, as pointed out in [31] the dynamic community detection methods discussed so far have a common limitation that communities and their evolution are studied separately. In this regard, Cazabet et al. [5] propose a robust overlapping community detection



method `iLCD` which adapts an initially detected community structure to the changes occurring in a dynamic network. However, it only considers the addition of new edges and nodes to the network and not the removal and only identifies the Merge, Growth and Birth of communities. Similarly, Nguyen et al. [17] (`AFOCS`), and Bhat and Abulaish [3] also aim to adapt a previous community structure to the dynamic changes in a network including removal. It should be noted that in this paper, we do not aim to deal with the dynamic nature of social networks and communities, instead we aim to propose an efficient parameter free density-based overlapping community detection method and compare it with the current state-of-the-art methods using rigorous experimental evaluations on a number of benchmark social networks.

### 3 PROPOSED METHOD

Along the lines of `SCAN` [60], `DENGRAPH` [14], and other recent density-based community detection methods like `GSK` [48], and `CHRONICLE` [22], the proposed `OCMiner` is based on `DBSCAN` [13] method where a cluster is searched by detecting the neighborhood of each object in the underlying database. The neighborhood of an object  $p$  is based on the similarity or inversely the distance between  $p$  and other objects in the underlying database. An object  $q$  belongs to the neighborhood of an object  $p$  if the distance between  $p$  and  $q$  is less than or equal to a threshold  $\varepsilon$ . In a graph-based context, a node  $q$  belongs to the neighborhood of a directly connected node  $p$  if the distance (structural) between  $p$  and  $q$  is less than or equal to  $\varepsilon$ . If the neighborhood of a given radius  $\varepsilon$  of a point  $p$  contains more than  $\mu$  objects, a new cluster with  $p$  as a core object is created. The process then iterates to find density-reachable objects from these core objects and defines a density-connected cluster using density-connectivity relations between nodes [13]. However, as pointed out in section 2, the main drawback of the traditional density-based community detection methods is their requirement of a global neighborhood threshold,  $\varepsilon$ , and minimum cluster size threshold,  $\mu$ , to be specified by the users. It would be more appropriate to somehow dissolve the effect of these parameters or at least the effect of  $\varepsilon$  on community structures discovered through density-based community detection methods to make them more flexible. A similar attempt has been made in [19], and along this direction, `OCMiner` follows a density-based approach for detecting overlapping community structures in social networks. The proposed method does not require the global neighborhood threshold parameter  $\varepsilon$  to be set manually at the beginning of the process. Instead, it uses a local representation of the neighborhood threshold which is automatically calculated for each node locally from the underlying social network using a much simpler approach. Moreover, a local version of  $\mu$  is also computed for each node automatically using a global threshold  $\eta$ . The proposed method thus requires only a single tunable parameter  $\eta$  to be set by the users.

#### 3.1 PRELIMINARIES

This section presents the formal definition of a novel distance function which has been used in `OCMiner` to determine the similarity of a node with its neighboring nodes in the network. It also discusses some basic concepts related to the development of `OCMiner`. For simplicity, a set of notations has been used. Table 1 presents the symbols and their brief descriptions. It should be noted that the proposed method is designed to work on directed and weighted networks wherein the weight of an edge is considered as the intensity of interactions (of any type) between the connected nodes. However, the method is generic and can also be applied on un-weighted and un-directed networks through assigning a unit weight to each edge and

considering each edge as a bidirectional edge having same weight. Mathematically, the interaction graph of a social network is defined as  $G_I = (V, E_w)$ , where  $V$  is the set of nodes and  $E_w \subseteq V \times V$  is the set of weighted links between nodes. For un-weighted networks each link can be assigned a unit weight value without altering the structural characteristics of the network.

Table 1: Notations and their descriptions

Notation	Description
$I_{\vec{p}}$	Total number of out-going interactions of a node $p$ (sum of the weights of all the outgoing edges from $p$ )
$I_{\vec{pq}}$	Number of interactions from node $p$ to node $q$ (weight of the edge from $p$ to $q$ )
$I_{\vec{pq}}$	Number of reciprocated interactions (weight) between $p$ and $q$ : $\min(I_{\vec{pq}}, I_{\vec{qp}})$
$I_{\vec{p}}$	Number of reciprocated interactions of a node $p$ : $\sum_{\forall q \in V_p} \min(I_{\vec{pq}}, I_{\vec{qp}})$
$V_p$	Set of nodes with whom node $p$ interacts
$V_{pq}$	Set of nodes with whom both nodes $p$ and $q$ interact: $V_p \cap V_q$

As mentioned earlier, an important component of density-based community detection methods is the similarity/distance function used to decide whether a pair of nodes can belong to the same community or not. For the proposed method, distance between two nodes is based on the average number of their reciprocated interactions and their commonly interacted nodes. More specifically, if  $p$  and  $q$  are interacting nodes and  $V_{pq}$  is the set of nodes with whom both  $p$  and  $q$  interact, then the similarity between  $p$  and  $q$  can be determined using the amount of response from  $p$  to  $q$  and to the nodes in  $V_{pq}$  as well as the amount of response from  $q$  to  $p$  and to the nodes in  $V_{pq}$ . The intuition here is that a pair of nodes whose interaction reciprocity with each other and with a set of commonly interacted nodes in the network is higher than the surrounding nodes (topological neighborhood) can be considered to be more related/close to each other than the pair which does not show such a behavior. Therefore, distance function is formulated in terms of "response", which is defined as follows:

**Definition 3.1 (Response)** For a pair of interacting nodes  $p, q \in V$ , response of node  $q$  to the interactions of node  $p$  is represented as  $\rho(p, q)$  and defined as the average of the per-user reciprocated interactions (link weights) of  $q$  and the nodes of  $V_{pq}$  with  $p$ , if  $I_{\vec{pq}} > 0$ , otherwise 0. Mathematically, it can be defined using equation 1, where  $V_{pq}$  and  $I_{\vec{pq}}$  have same interpretations as given in table 1.

$$\rho(p, q) = \begin{cases} \frac{\sum_{s \in V_{pq}} (I_{\vec{ps}}) + I_{\vec{pq}}}{|V_{pq}| + 1} & \text{if } I_{\vec{pq}} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The function  $\rho(p, q)$  gives the measure of a per-receiver average response (average reciprocated interactions) of a node  $q$  and the set of common receivers of  $p$  and  $q$  to the interactions of node  $p$ . In equation 1, a higher response from a node  $q$  to a node  $p$  represents more activity and hence more closeness of the two nodes with each other. It is obvious from the definition that  $\rho(p, q)$  is an asymmetric function. However, instead of defining an asymmetric directed response from a node  $q$  to a node  $p$  or vice versa and to ensure *determinism*



in the detection process, responsiveness between two nodes  $p$  and  $q$  can be generally taken as the maximum of their mutual directed-response, i.e.,  $\max\{\rho(p, q), \rho(q, p)\}$  or alternatively  $\min\{\rho(p, q)^{-1}, \rho(q, p)^{-1}\}$

**Definition 3.2 (Distance)** *Distance between a pair of nodes  $p, q \in V$  is represented as  $\Delta(p, q)$  and defined as the minimum of the reciprocals of their respective mutual directed-response values normalized by their respective total number of outgoing interactions in the interaction graph, provided  $\rho(p, q) > 0$  and  $\rho(q, p) > 0$ , otherwise it is assigned a value 1, as given in equation 2.*

$$\Delta(p, q) = \begin{cases} \min \left\{ \frac{\rho(p, q)^{-1}}{I_{\vec{p}}}, \frac{\rho(q, p)^{-1}}{I_{\vec{q}}} \right\} & \text{if } \rho(p, q) > 0 \wedge \rho(q, p) > 0 \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

In simple terms, the distance function defined in equation 2 gives the minimum *mutual reciprocal-response* between two nodes  $p$  and  $q$  in a social network. To normalize the distance in the range of  $[0, 1]$ , mutual reciprocal-response scores of both nodes  $p$  and  $q$  have been divided by their respective total count of outgoing interactions  $I_{\vec{p}}$  and  $I_{\vec{q}}$ , respectively. Smaller values of  $\Delta(p, q)$  represent higher response and thereby more closeness between them.

Generally, density-based community detection methods estimate neighborhoods of nodes to mark the close neighbors of each node  $p$  out of the nodes connected with  $p$  based on their distance from  $p$ . For this purpose, they use a manually determined global neighborhood threshold,  $\varepsilon$ , in such a way that a node  $q$  is assigned to the neighborhood of a node  $p$  only if the distance between  $p$  and  $q$  is less than or equal to  $\varepsilon$ . Instead of manually setting a value for  $\varepsilon$ , we propose a function to automatically determine a local version of the neighborhood threshold, termed as *local-neighborhood threshold*, for every node  $p$  from the underlying network itself.

**Definition 3.3 (Local-neighborhood threshold)** *For a node  $p \in V$ , the local-neighborhood threshold for  $p$  is represented by  $\varepsilon_p$  and defined as the average per-receiver reciprocated interaction score of  $p$  with all its neighbors, i.e., nodes with whom it has out-links. Formally,  $\varepsilon_p$  can be defined using equation 3, where  $\frac{I_{\leftarrow p}}{|V_p|}$  represents the average number of reciprocated interactions between the node  $p$  and all other nodes in  $V$  to whom it sends interactions, and the denominator  $I_{\vec{p}}$  represents the total count of outgoing interactions from node  $p$  and normalizes the value of  $\varepsilon_p$  in the range of  $[0, 1]$ .*

$$\varepsilon_p = \begin{cases} \frac{\left( \frac{I_{\leftarrow p}}{|V_p|} \right)^{-1}}{I_{\vec{p}}} & \text{if } |V_p| > 0 \wedge I_{\leftarrow p} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Based on the distance function and local-neighborhood threshold discussed above, a local version of  $\varepsilon$ -neighborhood, termed as *local  $\varepsilon$ -neighborhood* is defined as follows:

**Definition 3.4 (Local  $\varepsilon$ -neighborhood)** *The local  $\varepsilon$ -neighborhood of a node  $p \in V$  is represented by  $N^\varepsilon(p)$  and defined as the set of nodes to whom  $p$  sends interactions and their distance from  $p$  (i.e.,  $\text{dist}(p, q)$  equation 6) is less than or equal to  $\varepsilon_p$ , as shown in equation 4.*

$$N^\varepsilon(p) = \{q : q \in V_p \wedge \text{dist}(p, q) \leq \varepsilon_p\} \quad (4)$$

For the proposed method, a local version of minimum-number-of-points threshold is also computed automatically from the underlying network. However, it requires a *global threshold*,  $\eta$ , to be provided by the users as a fraction value, i.e.,  $0 < \eta \leq 1$ .

**Definition 3.5 (Local minimum-number-of-points threshold)** *For a node  $p \in V$ , the local minimum-number-of-points threshold is represented by  $\mu_p$  and taken as the the number of nodes specified by the fraction  $\eta$  of nodes to whom it has out-going interactions. Given a value of  $\eta$ , the value of  $\mu_p$  for a node  $p$  can be determined using equation 5.*

$$\mu_p = \eta \times |V_p| \quad (5)$$

It should be noted that the fraction  $\eta$  forms the only global parameter for the proposed method to be set by the users. Besides determining local minimum-number-of-points threshold values,  $\eta$  is also used to set a constraint, specified below, on the distance between two nodes while determining the local  $\varepsilon$ -neighborhood for a node  $p$ , i.e.,  $N^\varepsilon(p)$ .

**Constraint 3.6 (Distance constraint)** *Distance constraint specifies that the distance between two interacting nodes  $p$  and  $q$  can be measured by equation 2 only if the number of commonly interacted nodes of  $p$  and  $q$  is greater than the number of nodes specified by the fraction  $\eta$  of the minimum of their respective interacted nodes, minus one. Otherwise, distance between them is taken as 1. Formally, distance constraint can be formalized as shown in equation 6.*

$$dist(p, q) = \begin{cases} \Delta(p, q) & \text{if } |V_{pq}| > (\eta \times \min(|V_p|, |V_q|)) - 1 \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

**Definition 3.7 (Core node)** *A node  $p \in V$  having non-zero reciprocated interactions with other node(s) is defined to be a core node with respect to a global percentage parameter,  $\eta$ , if its local  $\varepsilon$ -neighborhood,  $N^\varepsilon(p)$ , contains at least  $\mu_p$  (local minimum-number-of-points threshold for  $p$ ) of its interacted nodes, as shown in equation 7.*

$$CORE^\eta(p) \Leftrightarrow |N^\varepsilon(p)| \geq \mu_p \quad (7)$$

The concept of *core node* is important for defining community in a density-based context. The proposed method uses the concept of core nodes to grow communities in a recursive manner based on the following definitions.

**Definition 3.8 (Direct density-reachability)** *A node  $q$  is direct density-reachable from a node  $p$  with respect to a global percentage threshold,  $\eta$ , if  $p$  is a core node and  $q$  belongs to the local  $\varepsilon$ -neighborhood of  $p$ , as shown is equation 8.*

$$DirREACH^\eta(p, q) \Leftrightarrow CORE^\eta(p) \wedge q \in N^\varepsilon(p) \quad (8)$$

Direct density-reachability is an asymmetric relation, i.e., if a node  $q$  is direct density-reachable from a node  $p$ , then it is not necessarily true otherwise.

**Definition 3.9 (Mutual-cores)** *Two nodes  $p$  and  $q$  are said to be mutual-cores if both  $p$  and  $q$  are core nodes, and  $p$  belongs to the local  $\varepsilon$ -neighborhood of  $q$ , and  $q$  belongs to the local  $\varepsilon$ -neighborhood of  $p$ . In other words,  $p$  and  $q$  are mutual-cores if  $p$  and  $q$  are direct density-reachable from each other, as shown in equation 9.*

$$\begin{aligned} MutCORES^\eta(p, q) \Leftrightarrow \\ DirREACH^\eta(p, q) \wedge DirREACH^\eta(q, p) \end{aligned} \quad (9)$$

**Definition 3.10 (Density-reachability)** *A node  $q$  is density-reachable from a node  $p$  with respect to a global percentage parameter,  $\eta$ , if there is a chain of nodes  $v_1, v_2, \dots, v_n$ , where  $v_1 = p$  and  $v_n = q$ , such that  $v_{i+1}$  and  $v_i$  are mutual-cores for  $i = 1, 2, \dots, n - 2$ , and  $v_n$  is direct density-reachable from  $v_{n-1}$ , as shown in equation 10.*

$$\begin{aligned} DenREACH^\eta(p, q) \Leftrightarrow \\ \exists v_1, \dots, v_n \in V : v_1 = p \wedge v_n = q \wedge \\ \forall i \in \{1, 2, \dots, n - 2\} MutCORES^\eta(v_i, v_{i+1}) \wedge DirREACH^\eta(v_{n-1}, v_n) \end{aligned} \quad (10)$$

Density-reachability is asymmetric and transitive, and it is not necessary that two nodes belonging to the same community will be density-reachable. They may belong to the same community because they are density-reachable through some other nodes belonging to that community. This condition is formalized in the following definition of density-connectivity.

**Definition 3.11 (Density-connectivity)** *A node  $q$  is density-connected to a node  $p$  with respect to a global percentage parameter,  $\eta$ , if there exists a node  $r$  such that both  $p$  and  $q$  are density-reachable from  $r$ , as shown in equation 11.*

$$\begin{aligned} DenCONNECT^\eta(p, q) \Leftrightarrow \\ \exists r \in V : DenREACH^\eta(r, p) \wedge DenREACH^\eta(r, q) \end{aligned} \quad (11)$$

Density-connectivity is a symmetric relation and for density-reachable vertices, it is also reflexive.

**Definition 3.12 (Density-connected community)** *A non-empty set  $C \subseteq V$  is said to represent a density-connected community with respect to a global percentage parameter,  $\eta$ , if all the vertices in  $C$  are density-connected with each other, and  $C$  is maximal with respect to density-reachability, as given in equation 12*

$$\begin{aligned} COMMUNITY^\eta(C) \Leftrightarrow \\ 1. \text{Connectivity} : \forall p, q \in C : DenCONNECT^\eta(p, q) \\ 2. \text{Maximality} : \forall p, q \in V : p \in C \wedge DenREACH^\eta(p, q) \Rightarrow q \in C \end{aligned} \quad (12)$$

Large real-world social networks are often found to contain *noise* or *outliers*, i.e., nodes that do not belong to any community, and *hubs*, i.e., nodes that do not belong to a particular community but connect multiple communities and thus play an important role in information brokerage and diffusion within a network and across communities. After finding all possible density-connected communities from a network, `OCMiner` considers the following definition to identify hubs and outliers in the networks.

**Definition 3.13 (Hubs and Outliers)** *An un-clustered node is considered as a hub if it has out-going links to the primary-core nodes of more than one community thus connecting the primary-communities of the respective core-nodes. Remaining un-clustered nodes that do not qualify as hubs are treated as outliers.*

In this paper we do not aim to deal with hubs explicitly, rather all un-clustered nodes are considered as outliers, unless explicitly specified.

### 3.2 COMMUNITY DETECTION PROCESS

This section presents the procedural detail of the proposed overlapping community detection method – **OCCMiner**. Initially, all nodes of a social network are un-labeled and marked as un-visited. For a given value of the global percentage threshold,  $\eta$ , community detection process iteratively finds a density-connected community by randomly selecting an un-visited node, say  $p$ , to grow a community using density-reachable relationship of  $p$  with other nodes in the network. For each un-visited node  $p$ , it checks whether  $p$  is a core node and if  $p$  qualifies the test, it finds all density-reachable nodes of  $p$  to identify its community. To do so, it first computes a local-neighborhood threshold for  $p$ ,  $\varepsilon_p$ , using equation 3. If  $\varepsilon_p$  is greater than zero, then it uses the distance constraint specified in equation 6 and the distance function of equation 2 to determine a local  $\varepsilon$ -neighborhood of  $p$ ,  $N^\varepsilon(p)$ . If node  $p$  qualifies as a core node, its community list is appended with the current community label and the community list of each node in  $N^\varepsilon(p)$  is also appended with the same. We use the term appended as the nodes belonging to  $N^\varepsilon(p)$  including  $p$  can already be labeled by some other community label(s) in some previous iteration(s). A node is assigned to a new community irrespective of its previous community allotments, thus allowing a node to belong to multiple communities. Once a node  $p$  is identified as a core-node, the following important steps are performed to identify a density-connected community around it.

1. All un-visited mutual-core nodes of  $p$  in  $N^\varepsilon(p)$  are appended with the current community label. They are marked as visited and pushed to a stack to identify the density-reachable nodes of  $p$ . This step is later repeated for each node in the stack for the current community to find the connected sequences of mutual-core nodes starting from  $p$ . These connected sequences of mutual-core nodes form the *Mutual-core Connected Maximal Sub-graph* (MCMS) of a community. All nodes in the MCMS of a community are called the primary-core nodes of that community. However, if a core-node  $p$  does not show mutual-core relation with any other core-node, then only the node  $p$  along with its  $N^\varepsilon(p)$  forms a community with  $p$  being its only primary-core node.
2. If a core-node  $q$  in  $N^\varepsilon(p)$  is not a mutual-core of  $p$ , it is appended with the current community label, however it is not pushed into the stack to grow the current community and its visited/un-visited status is kept unaltered. Being a core-node,  $q$  may have been pushed to the stack in some previous iteration based on its mutual-core relation with some primary-core node (other than  $p$ ) of the current community or some other community. In this case, the status of node  $q$  will currently be visited. Alternatively, it may be pushed to the stack in some later iteration based on its mutual-core relation with a primary-core node (other than  $p$ ) in the current community or some other community. In this case, the status of node  $q$  will currently be un-visited.

3. Non-core nodes in  $N^\epsilon(p)$  are marked as visited and appended with the current community label. Such nodes form boundary nodes for the community of  $p$  and are not pushed into the stack as they cannot be used to grow a community.

---

**Algorithm 1:** OCMiner( $G_I = (V, E_w), \eta$ )

---

```

/*  $G_I = (V, E_w)$  is a social network with set  $V$  of nodes and set  $E_w$  of weighted edges
*/
/*  $\eta$  is the resolution threshold at which the community structure is to be identified
*/
1 begin
2   foreach un-visited  $p \in V$  do
3     /* select an un-visited node  $p$  */
4      $currentID \leftarrow newCluster()$ ; /* generate a new community ID */
5      $N_{(p)}^\epsilon \leftarrow \{q \in V | DirREACH_\eta(p, q)\}$ ; /* determine the local neighborhood of  $p$  */
6      $p.visited \leftarrow true$ ; /* mark  $p$  as visited */
7     /* check for the core property */
8     if  $CORE_\eta(p)$  then
9       repeat
10        /* repeat until the community cannot be further expanded */
11         $p.PrimaryCommunity \leftarrow currentID$ ; /* assign the current community ID to  $p$ 
12        as its primary community */
13         $p.CommunitySet.add(currentID)$ ; /* assign the current community ID to the
14        community membership set of  $p$  */
15         $p.visited \leftarrow true$ ; /* mark  $p$  as visited */
16        foreach  $q \in N_{(p)}^\epsilon$  do
17          /* select a node  $q$  from  $p$ 's local neighborhood */
18           $q.CommunitySet.add(currentID)$ ; /* assign the current community ID to
19          the community membership set of each node in  $p$ 's local neighborhood */
20          if not  $q.visited$  then
21             $N_{(q)}^\epsilon \leftarrow \{r \in V | DirREACH_\eta(q, r)\}$ ; /* determine the local neighborhood
22            of  $q$  */
23            if  $MutCORES_\eta(p, q)$  then
24               $push(q)$ ; /* push the mutual-cores of  $p$  to the stack for growing
25              the community */
26               $q.visited \leftarrow true$ ;
27            end
28          end
29        end
30      until ( $p \leftarrow pop()$ ) is empty;
31    end
32  end
33 end

```

---

The steps through 1 – 3 are repeated for each node in the stack thus identifying a density-connected community for each randomly selected un-visited node  $p$  in the social network. It is worthwhile to note that if a core-node  $q$ , assigned to a community  $C$ , does not show a mutual-core relation with any primary-core node of  $C$ , then  $q$  is called a secondary-core node of community  $C$  and  $C$  is called a secondary-community of

$q$ . Alternatively, if a core-node  $r$  is a primary-core node of a community  $C$  (i.e.,  $r$  belongs to the MCMS of  $C$ ) then community  $C$  is called the primary-community of  $r$ . The whole process is repeated for each un-visited node to find overlapping community structures in the network. On completion of the above process, the set of labels assigned to a node represents with the set of community IDs to which it belongs to and un-labeled nodes (if any) represent outlier nodes, i.e., they do not belong to any community as they do not show an interaction behavior that is similar to a sufficient number of nodes in the network. The pseudo-code shown in algorithm 1 presents the overlapping community finding process in a formal way.

## 4 OVERLAPPING COMMUNITIES

As mentioned earlier, **OCMiner** aims to identify overlapping community structures in a social network. It does so by allowing a node  $q$  to belong to the  $\varepsilon$ -neighborhood of a core-node  $p$  irrespective of  $q$ 's previous community assignments in a density-based context as discussed in section 3.2. Thus a node can belong to multiple communities representing a node where multiple communities overlap. In the proposed context, such a node can have one of the following properties.

- *A node  $q$  belongs to multiple communities but it is not a core-node.* It means that a non-core node  $q$  belongs to the local  $\varepsilon$ -neighborhood of respective primary-core nodes of multiple communities. Being a non-core node,  $q$  represents a boundary node of its assigned communities, indicating that its assigned communities overlap at their boundaries. For example, in figure 1, two communities  $C_1$  and  $C_2$  overlap at a non-core node 'F'. Node 'F' could thus be considered of having a similar participation with both the communities.

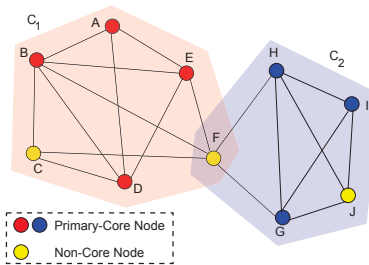


Figure 1: Overlapping communities sharing a non-core node 'F'

- *A node  $q$  belongs to multiple communities and is also a core-node.* It means that a primary-core node of one community is also a secondary-core for some other communities. This allows a community to overlap at a central node with other communities besides overlapping at the boundary. For example, in figure 2, two communities  $C_1$  and  $C_2$  overlap at a node 'J' which is a primary-core of community  $C_1$ . However, as the core-node 'J' also belongs to community  $C_2$  and does not show a mutual-core relation with any primary-core of community  $C_2$ , it forms the secondary-core of  $C_2$ .

It should be noted that for a community structure identified by **OCMiner** on a particular network, a community can have multiple primary-core nodes in its MCMS, but a core-node can be a primary-core node of only one community. This is unlike the method proposed in [12], where an overlapping node can be a central node



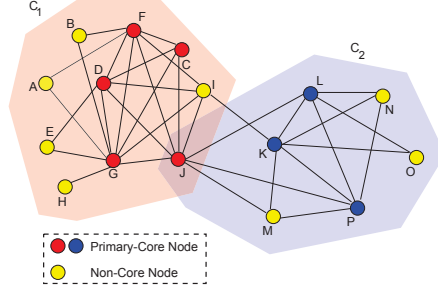


Figure 2: Overlapping communities sharing a core-node where the shared node ‘J’ is a primary-core of community  $C_1$  and a secondary-core of community  $C_2$

of more than one community. Moreover, it is also unlike the method proposed in [27], where overlapping nodes usually lie at the boundary of communities, whereas in the real-world networks they often are central nodes of a community. For example, considering the associations between the dictionary words as a network,

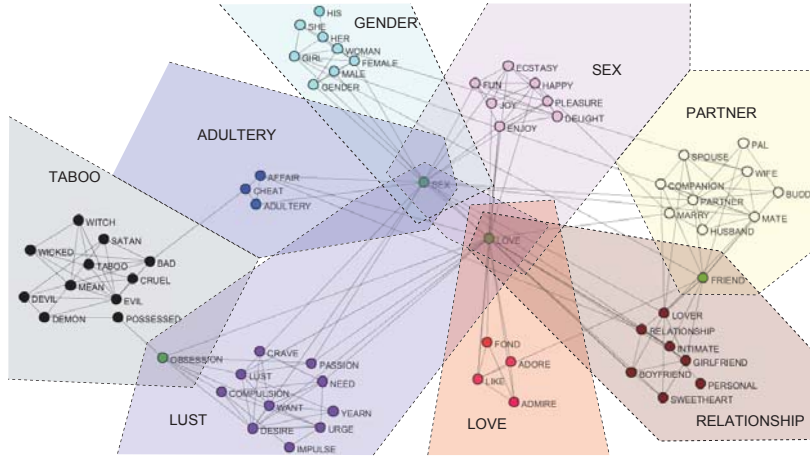


Figure 3: Partial community structure detected by **OCMiner** from a word association network

the words *sex* and *love* form central nodes of their respective communities but they form boundary nodes of other communities like *gender*, *adultery*, *lust* and so on. Figure 3 shows the overlapping community structures discovered by **OCMiner** on the un-weighted word association network [36] around the words *love*, *sex*, *partner*, and *taboo*. The figure implies that **OCMiner** significantly identifies the overlapping characteristics of the central nodes of the communities in the word association network. Using the proposed method, if a node  $p$  belongs to multiple communities, we can say that some primary-core nodes of the assigned overlapping communities of  $p$  show a similar interaction behavior with  $p$  as they show with other nodes in their respective local  $\varepsilon$ -neighborhoods.

It is often possible that two communities could overlap in such a way that majority of nodes (more than 50%) of one community (in some cases both the communities) are involved in the overlap between the two communities. In such cases two overlapping communities can be merged to represent a single community as implemented by [12]. In order to maintain uniformity for **OCMiner**, we reuse the threshold  $\eta$  to determine the merging criteria as follows. After the main community detection process is completed, **OCMiner** merges

two overlapping communities if the number of nodes, involved in the overlap between them, for the smaller community  $C$  is more than or equal to the number of nodes specified by the fraction  $\eta$  of  $C$ 's candidate nodes. For **OCMiner** this process is termed as *post-merge* and is applied after the main community detection process (section 3.2) is completed. Moreover, it should be noted that *post-merge* is applied in all the experiments performed in this paper.

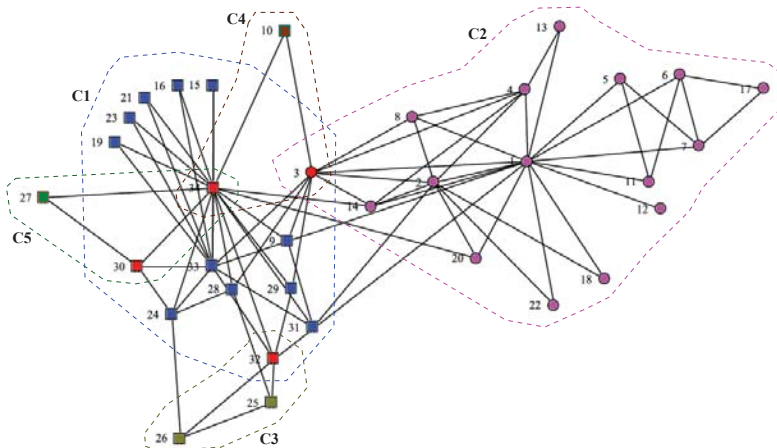


Figure 4: Result of **OCMiner** without *post-merge* (at  $\eta = 62\%$ ) on Zachary’s network showing five overlapping communities where communities  $C_4$  and  $C_5$  have a higher percentage of overlap with community  $C_1$

The result of **OCMiner** without using *post-merge* on weighted Zachary’s karate club network [61] identifies five overlapping communities at  $\eta = 62\%$  as shown in figure 4. In figure 4 the ground truth of the split of the network into two communities is shown by the shape of nodes (squares and circles). The communities identified by **OCMiner**, labeled from  $C_1$  –  $C_5$ , are represented by dashed boundaries and node colors where red color represents overlapping nodes. It can be seen in figure 4 that communities  $C_4$  and  $C_5$ , which consist of three nodes each, have majority of their nodes (more than 65%) overlap with nodes of a larger community  $C_1$ . Thus, communities  $C_1$ ,  $C_4$  and  $C_5$  can be merged to form a single community, as shown in figure 5.

On analyzing the results in figure 5, it can be seen that the communities  $C_1$  and  $C_2$  (represented by dashed borders) identified by **OCMiner** almost perfectly match the ground truth (represented by node shape) of the Zachary’s network with only two nodes labeled 25 and 26 being assigned to a separate overlapping community  $C_3$ . Moreover, node labeled 3 is classified as an overlapping node for the two main communities identified by **OCMiner**. It means that the whole group could be thought of being held together by node 3. Analogously, a dispute between nodes labeled as 1 and 33 had resulted in the actual split of the club. Moreover, node 3 is the only common neighbor of nodes 1 and 33 that also has highest number of neighbors in both the actual communities. Thus, it can be concluded that the communities identified by **OCMiner** on the Zachary’s network are meaningful and realistic.

## 5 EXPERIMENTAL RESULTS

This section presents the experimental results of **OCMiner** on many benchmark datasets including both real-world and synthetic social networks. It also presents a comparison of **OCMiner** with six other state-of-the-art

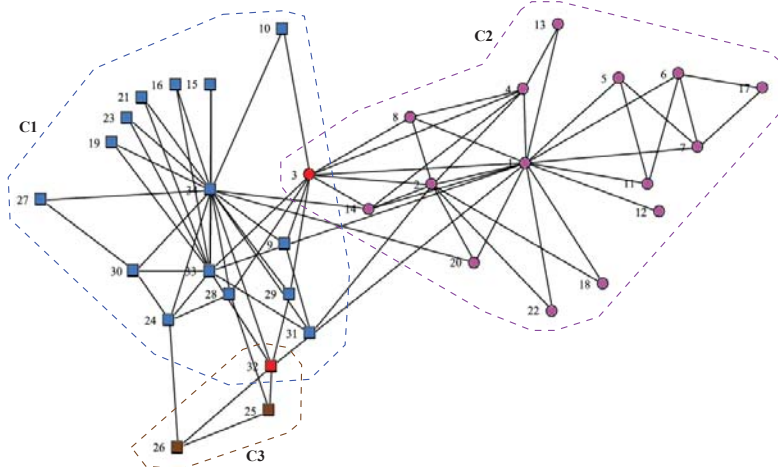


Figure 5: Result of *OCMiner* with *post-merge* (at  $\eta = 62\%$ ) on Zachary’s network, i.e., after merging highly overlapping communities

community detection methods, as mentioned in section 2, followed by a brief analysis of the obtained results. The scoring measures used to evaluate the community detection methods in question include – *O-NMI*, an extension of Normalized Mutual Information (NMI) [10], implemented by Lancichinetti et al. [27] and *Omega* [9], an extension of Adjusted Random Index (ARI) [20], both generalized for evaluating overlapping community structures. Further details about these measures are given in Appendix-1. While comparing two partitions using the scoring measures *Omega* and *O-NMI*, a score of 1 represents an exact match between the two partitions being compared. Thus for a community detection method to score better on these measures, it is expected to yield a value closer to 1. We also provide some information related to the accuracy of the identified overlapping communities and nodes by the various methods where required. The methods *SHRINK* [19] and *GSK* [48] have been shown to perform better than *SCAN* [60] in their respective experiments. *LFM*[27] and *CFinder* [38] (implementing CPM) perform better than *EAGLE* [45] in the experiments performed in [57]. So, we do not compare our method with *EAGLE* and *SCAN* (requiring density parameters  $\varepsilon$  and  $\mu$ ). It should also be noted that, *GSK* and *MOSES* [35] are parameter free methods and are not sensitive to any input parameter. On the other hand *CFinder* requires an input parameter  $k$  to define the clique size. However, [38, 26, 57] and our preliminary experiments confirm that *CFinder* yields best results at  $k = 4$ , and we use the same in the experiments presented in this paper. *LFM* requires an input value for the resolution parameter  $\alpha$ , but is shown to perform best at  $\alpha = 1$  in [27, 57], and hence we use the same in our experiments. For *SLPA* [59, 58] and *OCMiner*, we present the best results obtained by varying their respective input parameters; however, for *OCMiner*, we have devised a way to estimate an optimal value for its input parameter and presented the same later in this paper. All the experiments have been performed on an *INTEL*<sup>®</sup>*i3* based system with *4GBs* memory.

## 5.1 RESULTS ON REAL-WORLD NETWORKS

We have used six well-known static real-world network benchmarks to evaluate the performance of *OCMiner* and compare it with other state-of-the-art methods based on the *O-NMI* and *Omega* scores obtained as shown

in figure 6. The six networks include the Zachary’s network [61] which is a weighted interaction network between 34 members of a Karate club that split into two communities, the NCAA College football network [16] which is a social network consisting of 115 college football teams divided into eleven conferences and five independent teams, the Dolphin network [32] which is an un-directed and un-weighted social network of frequent associations between 62 Dolphins consisting of two communities, the US political books network is a network of 105 books about US politics<sup>1</sup> sold online by Amazon, a collaboration network [4] of 241 physicians, and a primary school interaction network of students [47]. For all six real-world networks, the ground truth community structures are known and are used to calculate the performance scores. Moreover, we also present information about the outliers and overlapping nodes detected by the methods in figure 7. It can be observed that the method `CFinder` does not produce any results for the primary school networks as this network is dense (5899 edges between 236 nodes) which increases the complexity for `CFinder`’s clique percolation approach. On the other hand, `OCMiner` finds community structures which closely match the ground truth for this network, indicating that it has no problems with networks containing dense cliques.

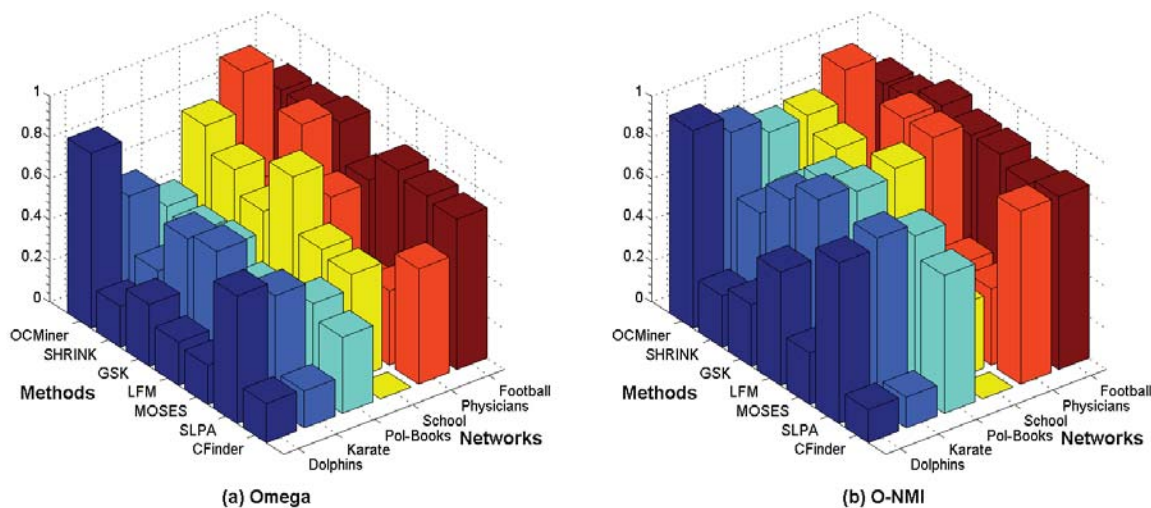


Figure 6: Experimental results on static real-world benchmarks.

As can be seen from figure 6 in general, for each of the six real world networks used, `OCMiner` performs better than all other methods in question on majority (four) of the networks and also performs comparable to the better performing methods `GSK` and `LFM` on other (two) networks. This is because `OCMiner` gets the `O-NMI` and `Omega` score closer to 1 for most of the networks used here and also provides a relatively better score as compared to the other methods in aggregate. In general, on the real-world benchmark networks,

## 5.2 RESULTS ON SYNTHETIC NETWORKS

Lancichinetti and Fortunato [26] have proposed a synthetic network generation method that can generate a class of artificial networks, usually referred to as LFR-benchmarks. They have claimed to reflect the important aspects of real networks and can be used as benchmarks for testing community detection algorithms. We have used their method to generate various synthetic networks for our experiments through varying various

<sup>1</sup><http://www.orgnet.com/>

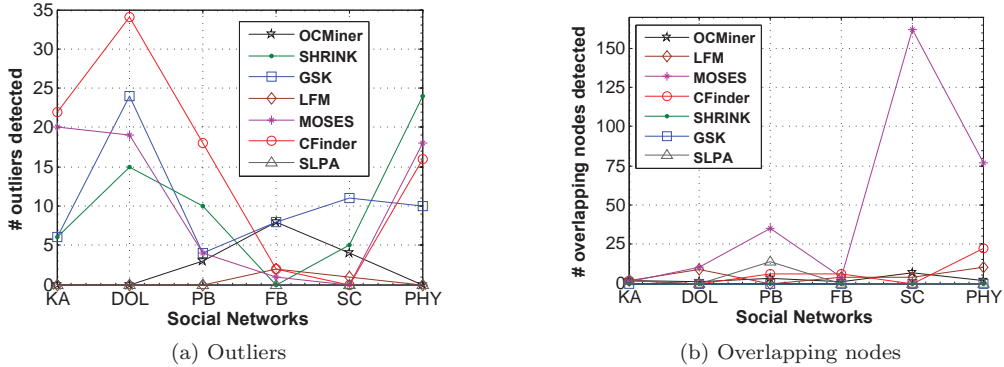


Figure 7: Outliers and overlapping nodes detected by the various methods on real-world networks

parameters required for the generation of networks. A description of the available parameters can be seen in their original paper [26]. Here, we only mention the important characteristics for each type of benchmark generated. For the synthetic networks generated, the number of nodes is set to 5000 and two different ranges for the community sizes  $S$ , 10–50 nodes (small) and 20–100 nodes (big) have been used. The average degree  $\langle k \rangle = 10$  and the maximum degree is  $k_{max} = 50$ , however for the directed networks the average degree  $\langle k \rangle = 20$  so as to simulate reciprocation of interactions between nodes as shown by real world interaction networks. The exponents of the degree and community size distributions for all LFR-benchmarks are set to  $\tau_1 = 2$  and  $\tau_2 = 1$ . Other special parameters and properties that are specific to different kinds of generated networks have been mentioned in the following sections. Moreover, each point of the resulting line graphs mentioned in the following sub-sections corresponds to an average over 50 realizations of the benchmark.

### 5.2.1 LFR-Benchmarks with Disjoint Communities

In this sub-section we aim to show that firstly, the proposed community detection method `OCMiner` identifies significant community structures even when the underlying network contains disjoint community communities. Secondly, we aim to compare the performance of `OCMiner` with the state-of-the-art parameter free density-based methods `GSK` and `SHRINK` that can use both jaccard based and cosine based similarity metrics. The metric used to generate the following results for these two methods was the one that yielded the best results, i.e., the cosine based measure.

To generate un-weighted and un-directed LFR-benchmarks, the topology mixing parameter,  $\mu_t$ , is varied between the range 0.1 – 0.6, with a step size of 0.05. Figure 8 gives a performance comparison of `OCMiner` with two other related density-based community detection methods – `GSK` and `SHRINK`, based on the `Omega` and `0-NMI` scores on the generated un-weighted and un-directed LFR-benchmarks with disjoint communities. The networks used in figures 8a and 8b have community size ranging between 10 – 50 nodes, whereas in figures 8c and 8d, the community size ranges between 20 – 100 nodes. The parameter  $\eta$  for `OCMiner` is varied between 25% – 35% to get the best results. Figures 8a and 8c show `Omega` scores, and figures 8b and 8d show the `0-NMI` scores. As the mixing parameter  $\mu_t$  increases, it becomes difficult to identify community structure; however, `OCMiner` performs better than `GSK` and `SHRINK` on this benchmark.

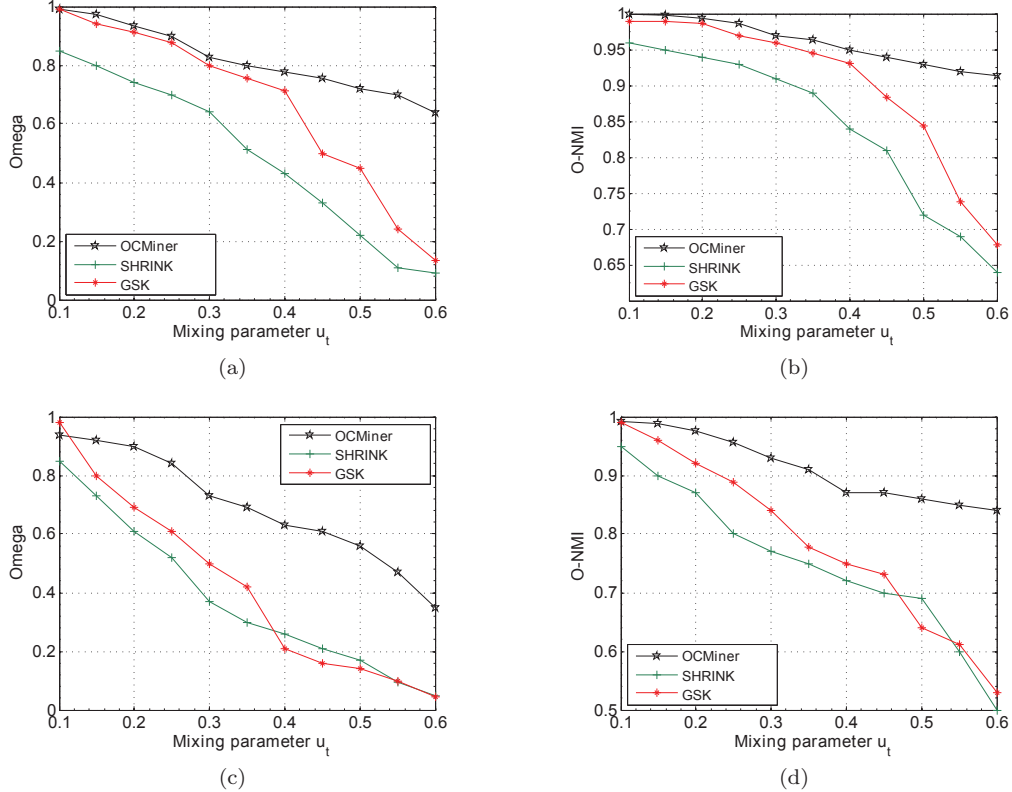


Figure 8: Experimental results on un-weighted and undirected LFR-benchmarks with disjoint communities

### 5.2.2 LFR-Benchmarks with Overlapping Communities

To evaluate the performance of **OCMiner** for detecting overlapping communities, LFR-benchmarks have been generated by varying the number of community memberships of an overlapping node from 3–8, and keeping the fraction of overlapping nodes fixed at 10%. Community size for all LFR-benchmarks with overlapping communities is relatively large, ranging between 20–100. It should be noted that for LFR-benchmarks with overlapping communities, density-based methods including **SHRINK** and **OCMiner** tend to label a majority of overlapping nodes as outliers. In fact, they actually qualify as hubs (nodes connecting multiple communities but belonging to none). **SHRINK** allows to mark a hub as an overlapping node by assigning it to each community which it connects. Similarly, **OCMiner** (besides identifying actual overlapping nodes) also finds hubs (as discussed earlier in section 3.1) and treats them as overlapping nodes. Here, **CFinder** is considered as it has shown to be out-performed on LFR-benchmarks by **SLPA** in [58]. Moreover, **GSK** is also not considered as it does not find overlapping communities. We compare the results based on the  $\Omega$  and O-NMI scores as the ground truth community structure for these networks is known. We also provide information on the characteristics of overlapping nodes detected by the various methods, using **FScore** accuracy measure given by equation 13.

$$FScore = \frac{2 * precision * recall}{precision + recall} \tag{13}$$



In equation 13, *precision* and *recall* relate to the fraction of true overlapping nodes out of the overlapping nodes detected and the fraction of true overlapping nodes detected as the same, respectively. An **FScore** value close to 1 for a community detection method means that the method shows higher accuracy in identifying the actual overlapping nodes of a network as the same. We also compare the ratio of the average memberships of the overlapping nodes detected in a particular network,  $O_m^d$ , by the community detection methods in question with the actual memberships of the overlapping nodes  $O_m$  in the respective networks. A value close to 1 for this ratio on a network means that the method closely identifies the actual memberships of the detected overlapping-nodes from the underlying network. We generate three types of networks and present the results as follows.

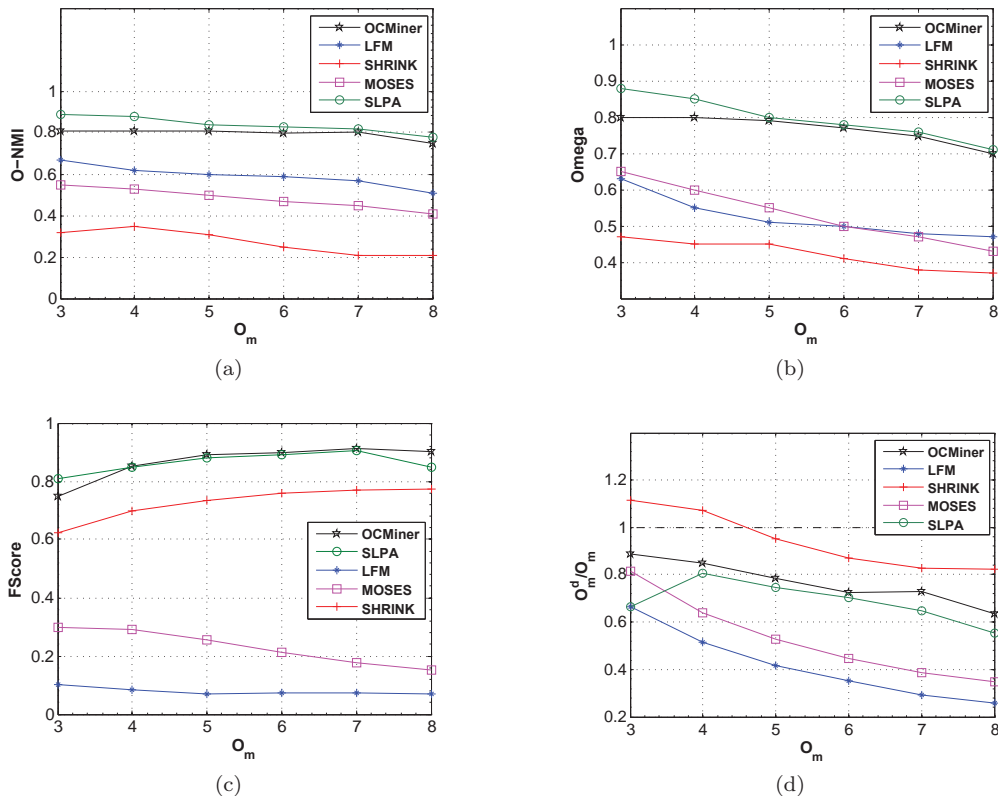


Figure 9: Experimental results on un-weighted and un-directed LFR-benchmarks with overlapping communities

**Un-directed and Un-weighted Networks:** For these LFR-benchmarks, we set the topology mixing parameter  $\mu_t = 0.1$ , and rest of the parameters are same as discussed earlier. Figure 9 shows a comparison of the results of various methods using four scoring measures mentioned earlier for overlapping communities. Figures 9a and 9b show that considering **O-NMI** and **Omega** scores, **SLPA** performs better than the other methods, however, results of **OCMiner** are comparable to that of **SLPA** for two measures. Figure 9c compares the **FScores** of various methods on un-directed and un-weighted benchmarks and shows that for **FScore**, **OCMiner** performs better than the other methods, however in this case, results of **SLPA** are comparable to that

of OCMiner. Figure 9d compares the ratio of the average membership of overlapping nodes detected by various methods with the actual memberships of the overlapping nodes,  $O_m^d/O_m$ , on different networks. It shows that SHRINK performs better on this measure (with the ratio close to 1) for the current benchmark than all other methods, followed by OCMiner and SLPA, respectively. However, considering all four scoring measures, it is OCMiner and SLPA that perform better than the other methods, and their results are comparable to each other on the current benchmark.

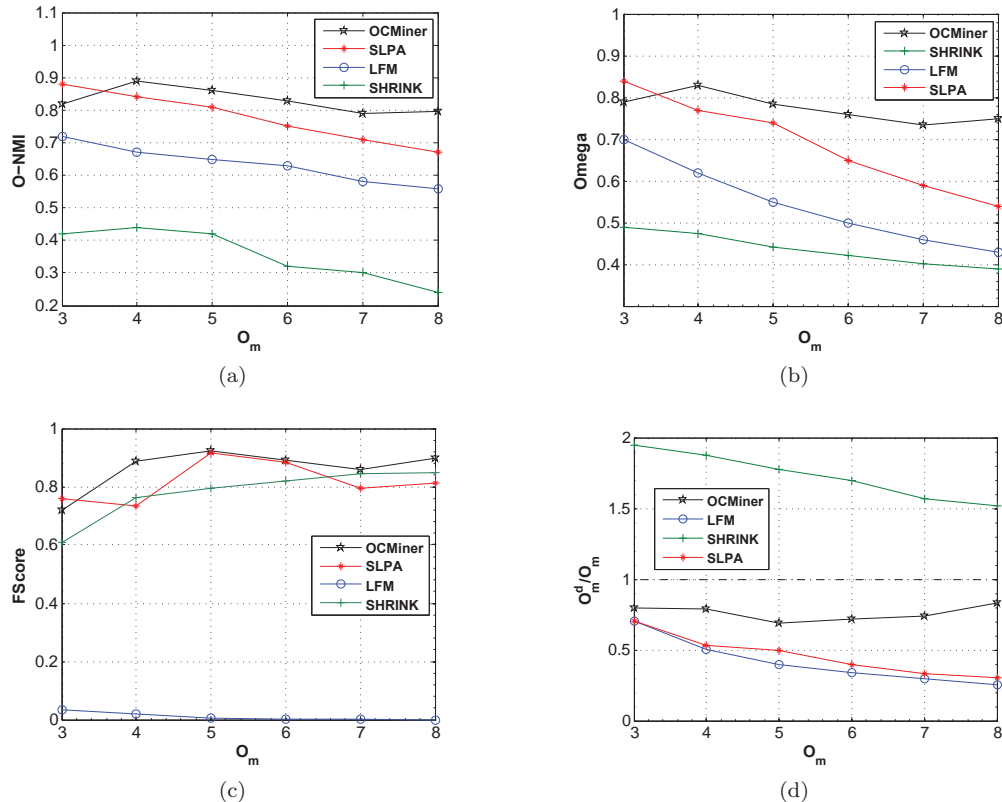


Figure 10: Experimental results on weighted and un-directed LFR-benchmarks with overlapping communities

**Un-directed and Weighted Networks:** For weighted LFR-benchmarks with overlapping communities, the mixing parameter for weights  $\mu_w$  and topology  $\mu_t$  have the relation  $\mu_w = \mu_t = 0.3$ , and rest of the parameters are same as discussed earlier. For this benchmark, we have not generated results for MOSES as it does not take links weights into consideration. Figure 10 shows a comparison of the results of various methods using all four scoring measures used in this paper for overlapping communities. It is clear from the figure that OCMiner performs better than all other methods for all four scoring measures, followed by SLPA.

**Directed and Weighted Networks:** For the case of weighted and directed LFR-benchmarks with overlapping communities, the mixing parameter for weights,  $\mu_w$ , and topology,  $\mu_t$ , have the relation  $\mu_w = \mu_t = 0.1$ , and rest of the parameters are set as discussed earlier. For this benchmark, we have generated results only for OCMiner and SLPA as among the methods in question only these two consider both directed and

weighted nature of networks. Figure 11 shows the comparison of the results of these two methods using four scoring measures. It can be observed from this figure that `OCMiner` performs better than `SLPA` on all scoring measures for the current LFR-benchmark. From the results obtained on the synthetic networks

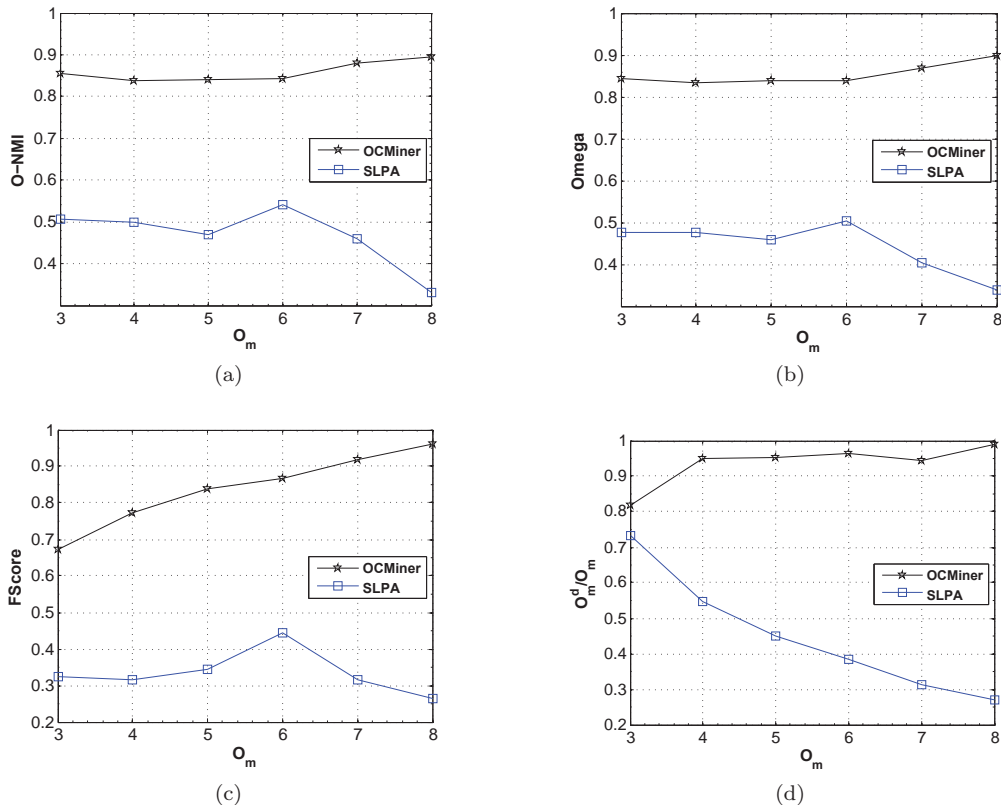


Figure 11: Experimental results on weighted and directed LFR-benchmarks with overlapping communities

with overlapping communities, in general, it can be concluded that `OCMiner` performs better than the other related methods, especially when the edge weights are available, i.e., when the networks are weighted.

### 5.3 RESULTS ON AMAZON CO-PURCHASE NETWORK

To evaluate the performance of `OCMiner` on a large social network, we have used the Amazon<sup>2</sup> co-purchase data which was collected on crawling Amazon website by [29]. It contains product metadata and review information about 548,552 different products (Books, music CDs, DVDs, and VHS video tapes). The products are assigned to various categories by Amazon and we consider the ones based on subjects(books), styles(Music), and genres(DVDs and VHS movies) resulting in a total of 13,684 highly overlapping product categories. We form a directed network between the products by creating a directed edge from a product to each of its co-purchased products. We filter out isolated nodes and nodes that have their in-degree or out-degree equal to 0. This results in a directed network of 234,083 nodes with 828,737 directed edges. Here we present results only for `OCMiner`, `CFinder` and `SLPA` as among the methods used in this paper,

<sup>2</sup>[www.amazon.com](http://www.amazon.com)

only these three consider the directed nature of edges in a network. Figure 12 shows the size distribution of the communities detected by the three methods on the Amazon co-purchase network wherein **OCMiner** finds 41,559 communities with 40,074 overlapping nodes, **CFinder** finds 22,613 communities with 11,241 overlapping nodes and **SLPA** finds 35,369 communities with 45,171 overlapping nodes. Figure 12 shows

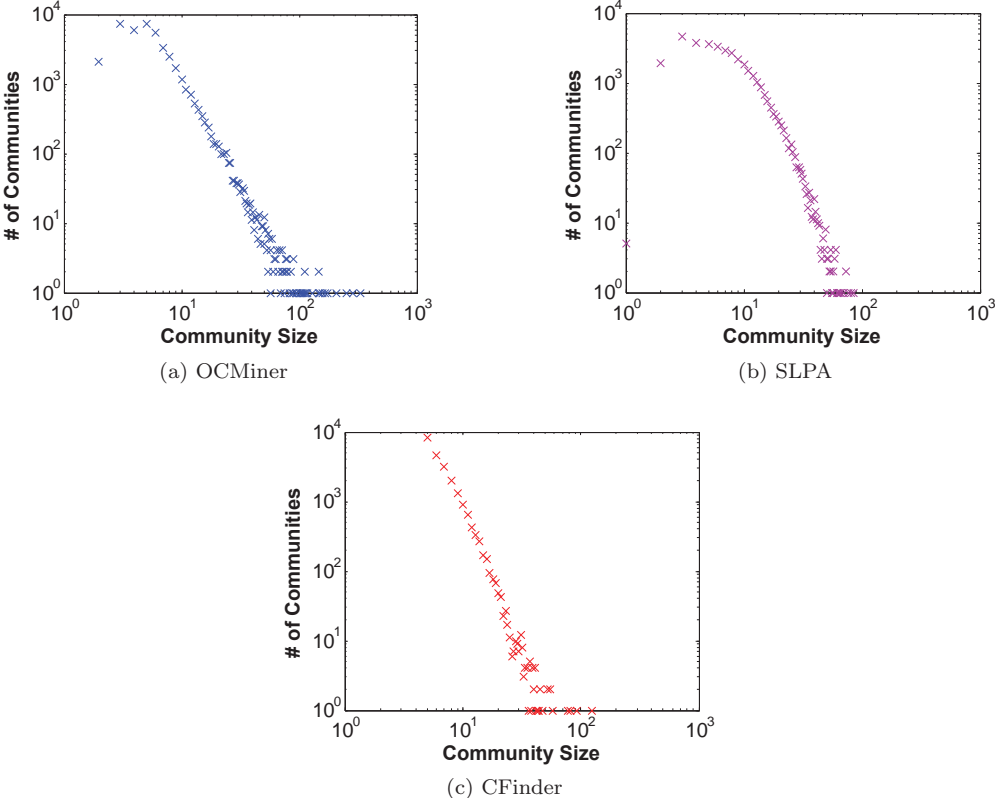


Figure 12: Size distribution of communities identified on directed and un-weighted Amazon product co-purchase network

that **CFinder** and **SLPA** tend to find communities with relatively smaller size (less than 100 nodes) while as **OCMiner** finds a significant number of communities with size greater than 100. Although, the nodes in the network are already grouped into categories, these categories cannot be considered as the ground truth community structure for the network as the co-purchase relations are not shown exclusively within the product-categories, but include a large number of cross-category relations. However, it is acceptable to consider that a community of products based on co-purchase relations can contain many products which represent a product-category or subset of a product-category. In this regard, for the Amazon network, we consider a community detected by a method as significant if more than 50% of its nodes form a subset of any product-category specified by Amazon. Figure 13a shows the fraction of nodes assigned to communities by the three methods and figure 13b shows the fraction of significant communities detected by the three methods on the Amazon co-purchase network. From 13a, it can be observed that **CFinder** marks 46% of the nodes as outliers, **OCMiner** marks 6% of nodes as outliers, and **SLPA** finds no outliers. **SLPA** aims to assign each node in the underlying network to a community and does not consider finding outliers. However,

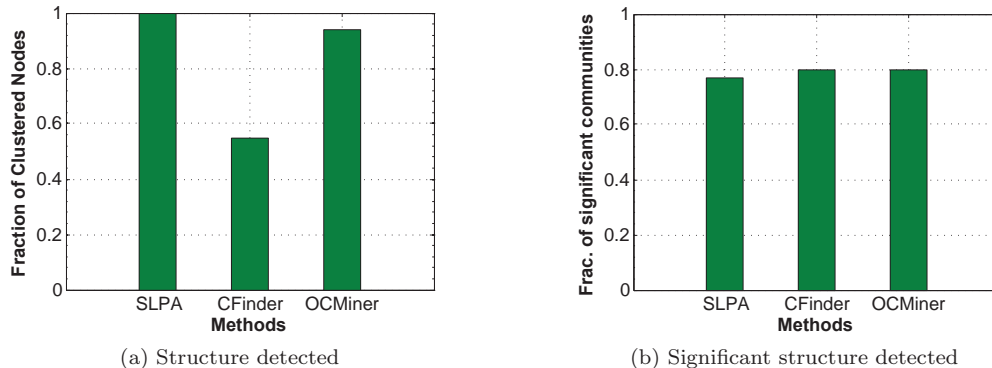


Figure 13: Amount of (a) structure, and (b) significant structure detected by SLPA, CFinder, and OCMiner on Amazon product co-purchase network

real-world networks often contain noise and outliers and identifying such nodes is often desirable. On the other hand CFinder marks too many nodes as outliers while OCMiner finds a small percentage of outliers. Based on the significance of the communities from figure 13b, for OCMiner and CFinder, 80% of the detected communities are significant which is more than 77% as detected by SLPA. Thus, considering both the amount of structure found in the network and the significance of communities detected by the three methods, it can be concluded that OCMiner and SLPA perform comparable. However, stressing on the fact that large real-world networks contain outliers and detecting them is often desirable, OCMiner performs better than both of them as CFinder marks too many nodes as outliers, whereas SLPA does not find outliers.

## 6 HEURISTIC FOR ESTIMATING $\eta$

Traditional density-based community detection methods like SCAN [60], DENGGRAPH [14], CHRONICLE and so on require two input parameters  $\varepsilon$  (*distance threshold*) and  $\mu$  (*minimum points*) with high sensitivity towards  $\varepsilon$ . OCMiner on the other hand requires only a single parameter,  $\eta$ , relating to the  $\mu$  (minimum points) of traditional methods, to be set by the users for detecting overlapping community structures in a social network. The value of  $\eta$  basically defines the size characteristic of the overlapping communities to be detected. Smaller values of  $\eta$  yield larger communities, whereas larger values yield smaller communities. It means that for OCMiner  $\eta$  can be tuned to detect overlapping community structures at different levels of size characteristics from social networks thus naturally forming a hierarchical representation of overlapping community structures. This feature assigns OCMiner to the multi-resolution class of hierarchical community detection methods. The demonstration of hierarchical overlapping community structure results obtained by OCMiner on the un-directed and un-weighted social network of frequent associations between 62 Dolphins in a community living off Doubtful Sound, New Zealand is shown in figure 14. From figure 14, it is clear that at  $\eta = 50\%$ , OCMiner exactly finds two communities that almost perfectly match the ground truth (represented by leaf-node shape and color in the dendrogram) of the Dolphin network. Node labeled as ‘PL’ is identified as an overlapping node between the two identified communities and node labeled as ‘SN100’ is the only node to be misclassified. Increasing the value for  $\eta$  from 50% to 60% breaks one of the two communities into three smaller communities thus resulting in a total of four communities with no outliers. Similarly, at  $\eta = 70\%$

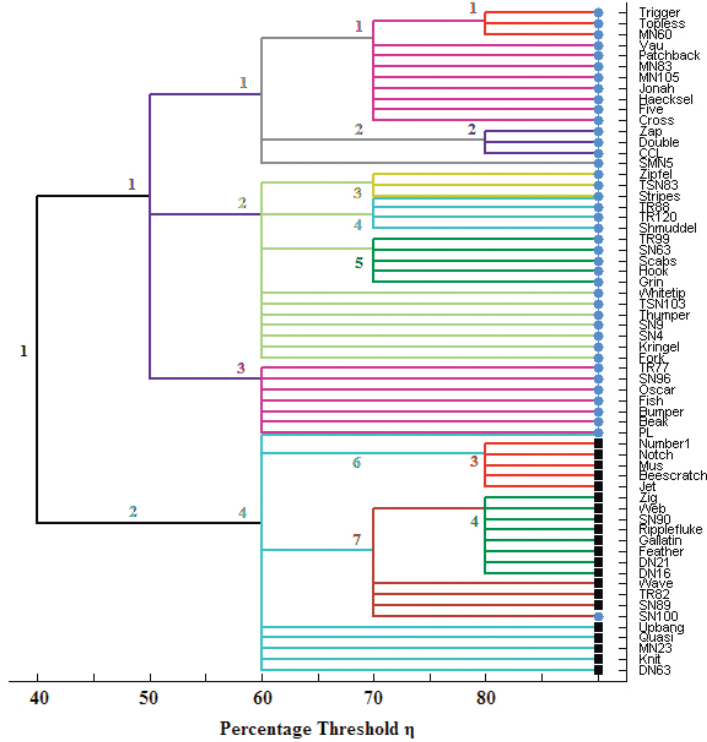


Figure 14: Color coded dendrogram representation of a hierarchical and overlapping community structure discovered by `OCMiner` from Dolphin network

`OCMiner` identifies seven smaller communities in the Dolphin network along with many isolated nodes which represent outliers.

However, besides hierarchical community structure, it is mostly required to determine community structure (at a single level) which is closest to the underlying community structure of a network. Thus for `OCMiner`, it is desirable to find an optimal value for  $\eta$  which reveals the best possible community structure for a network. Based on our experimental results on many real-world and synthetic networks, we observe that an optimal value of the input parameter  $\eta$  can be given by equation 15, where  $\Theta_{mean}$  represents the *mean* of the *topological overlap* (equation 14, where  $N_i$  represents the number neighbors of node  $i$  in the network) taken over all pairs of nodes between which there exists an edge in the underlying network (first rounded-up to two decimal places then truncated to one decimal place). And  $\Theta_{std}$  represents the *standard-deviation* of the topological overlap shown by the nodes around  $\Theta_{mean}$  in the network.

$$\Theta = \frac{|N_i \cap N_j|}{\min(|N_i|, |N_j|)} \quad (14)$$

$$\eta = \Theta_{mean} + \Theta_{std} \quad (15)$$

Table 2 compares the values of parameter  $\eta$  at which `OCMiner` shows the best results (preferred range) for each real-world network against the value estimated by equation 15 for the same network. From table 2 it can be seen that for all real-world networks used in this paper, the value of  $\eta$  determined for `OCMiner` using



Table 2: Significance of the values of  $\eta$  for **OCMiner** estimated using equation 15

Network	Parameter $\eta$	
	Preferred range	Estimated value
Karate	0.62-0.65	0.62
Football	0.5-0.6	0.54
Dolphin	0.5-0.55	0.5
Pol-books	0.6-0.65	0.61
School	0.6-0.66	0.64
Physicians	0.25-0.35	0.28
Amazon	0.4-0.45	0.41

equation 15 falls within the range of values at which **OCMiner** yields the best results. A Similar behavior was shown on the synthetic networks too. Based on these results, we claim that **OCMiner** can automatically find good approximations for its input parameter  $\eta$  and generate meaningful community structures from social networks. Moreover, by tuning the constant  $c$  in equation 15 with a step size of 0.15, a hierarchical community structure for a given network can also be generated.

## 7 COMPARISON OF RUNNING TIME

Considering the time complexity, the main part of **OCMiner** involves analyzing the local neighborhood of each node in the network, and for each node this cost is proportional to its out-degree. Thus the total cost for this step is  $O(deg(p_1) + deg(p_2) + \dots + deg(p_n))$ , where  $deg(p_i), i = 1, 2, \dots, n$  is the out-degree of each node  $p_i$  in the underlying network. For a complete graph of  $n$  nodes, the degree of each node is  $n - 1$ , leading to a worst case complexity for this step as  $O(n^2)$ . However, in general, real-world networks show sparser degree distributions, resulting in an  $O(n)$  average case complexity. In reality, **OCMiner** also involves a post-merge step (whose complexity depends on the number of identified overlapping communities with relatively many common/overlapping nodes) and the heuristics for estimating  $\eta$  (whose complexity depends on the number of edges in the networks). This makes it difficult to provide a true computational analysis of the method.

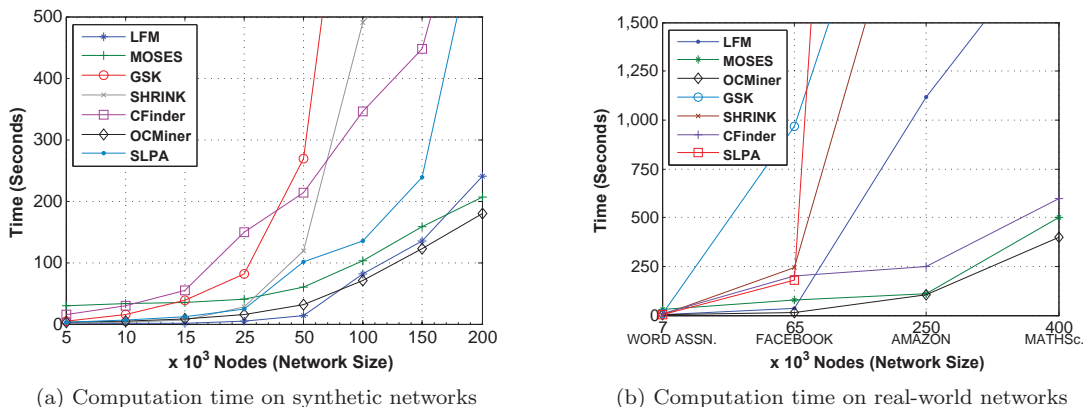


Figure 15: Computation time required by the various methods on different synthetic and real-world networks of varying sizes

In this regard, figure 15 compares the running time required by the various methods used in this paper to find community structure on a range of synthetic and real-world networks of different sizes to provide a rough computational view of the methods. The networks used for figure 15a are LFR-synthetic networks generated with the average degree  $\langle k \rangle = 10$  and maximum degree  $\langle k \rangle = 50$  by varying the number of nodes from 5,000 to 200,000. On the other hand, networks used for figure 15b represent some real-world networks which include the word association network [36] with more than 7,000 nodes, Facebook New Orleans friendship Network [51] with more than 60,000 nodes, Amazon co-purchase network [29] with approximately 250,000 nodes, and MathSciNet co-authorship network [39] with approximately 400,000 nodes. Figure 15a shows that for the range of synthetic networks, **LFM**, **OCMiner** and **MOSES** (to some extent) perform comparable to each other and better than the other methods. However, considering the large scale networks with more than 100,000 nodes, **OCMiner** performs better than the other methods. Similarly, results shown on the real-world networks (figure 15b) reveal that **OCMiner** performs faster than the other methods on both small and large networks. From these results, we conclude that **OCMiner** is faster than the state-of-the-art methods considered in this paper, and applicable to very large scale networks.

## 8 CONCLUSION AND FUTURE WORK

This paper has presented a novel density-based method, **OCMiner**, to identify overlapping community structures in social networks. Unlike other density-based methods for which the neighborhood threshold needs to be set by the users, the proposed method determines the neighborhood threshold for each node locally from the network itself. **OCMiner** is designed to detect communities in networks that represent important functional modules in the real-world networks like, protein-protein interaction modules and gene regulatory modules in biological networks, meaningful social groups in social networks like friendship circles, groups of people sharing common interests or activities, gang modules in criminal networks, work groups within organizations, and so on. **OCMiner** allows for communities to overlap which represents the fact that nodes in real networks can belong to multiple functional groups. It is suitable for detecting overlapping community structures and outliers in large-scale social networks. Our experimental results have shown that community structures identified by **OCMiner** on some of the well-known benchmark networks are significant and in general better than the state-of-the-art methods considered in this paper. **OCMiner** takes both the weighted and directed nature of networks into consideration besides finding community structures in un-weighted and un-directed networks. It means that **OCMiner** can find community structures in both un-weighted/un-directed friendship networks as well as directed/weighted interaction network in case of online social networks where information on topological community structures (based on friendship and/or interactions) can aid in activities like recommendations, etc. **OCMiner** also enables to explore the hierarchical community structure of a network by varying the input parameter  $\eta$ . Since the proposed method is relatively faster, it can also be used to analyze the dynamic nature of communities for analyzing the evolving friend and interaction relations of users by considering smaller time windows of an evolving network. We aim to work along this direction in addition to incorporating an evolutionary community visualization technique in the near future. Moreover, we also aim to provide an in-depth analysis related to the significance of edge weights of networks (if any) for the purpose of community detection as soon as related datasets are prepared or made available.

In [25], the authors have shown how communication reciprocity, communication interaction average, and

clustering coefficient of the nodes in online social networks can be used to differentiate spammers from normal users. The interactions of spammers are least often reciprocated and the communication interaction average of spammers is close to zero as most of the spam are simply ignored or discarded by recipients. Even if the recipient is interested in the subject described in a spam, the usual action is to follow a hyperlink in the spam instead of replying to the email. Furthermore, the neighbor accounts of spammers are unlikely to exhibit friends-of-friends relationships and thus show low clustering coefficients. As discussed earlier, `OCMiner` considers only reciprocated interactions of nodes in a social network. It takes the average interaction of a node  $p$  with its neighbors to determine its local neighborhood threshold and considers only those neighbors of  $p$  as its community members with whom its interaction behavior is same or better than the average. In this view, `OCMiner` can be useful to label spam nodes in online social networks among the detected outliers, and thus it can help at the initial phases of determining spammers' profiles in an online social network.

## References

- [1] G. Agarwal, D. Kempe, Modularity maximizing network communities using mathematical programming, *The European Physical Journal B* 66 (2008) 409–418.
- [2] S. Y. Bhat, M. Abulaish, A density-based approach for mining overlapping communities from social network interactions, in: *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, WIMS '12*, ACM, New York, NY, USA, 2012, pp. 9:1–9:7.
- [3] S. Y. Bhat, M. Abulaish, `OCTracker`: A density-based framework for tracking the evolution of overlapping communities in OSNs, in: *Advances in Social Networks Analysis and Mining (ASONAM)*, 2012 IEEE/ACM International Conference on, 2012, pp. 501–505.
- [4] R. S. Burt, Social contagion and innovation: Cohesion versus structural equivalence, *American Journal of Sociology* 92 (6) (1987) 1287–1335.
- [5] R. Cazabet, F. Amblard, C. Hanachi, Detection of overlapping communities in dynamical social networks, in: *Social Computing (SocialCom)*, 2010 IEEE Second International Conference on, 2010, pp. 309–314.
- [6] A. Clauset, Finding local community structure in networks, *Physical Review E* 72 (2005) 026132.
- [7] A. Clauset, C. Moore, M. E. Newman, Hierarchical structure and the prediction of missing links in networks, *Nature* 453 (7191) (2008) 98–101.
- [8] A. Clauset, M. E. Newman, C. Moore, Finding community structure in very large networks, *Physical Review E* 70 (2004) 066111.
- [9] L. M. Collins, C. W. Dent, Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions, *Multivariate Behavioral Research* 23 (2) (1988) 231–242.
- [10] L. Danon, A. Daz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, *Journal of Statistical Mechanics: Theory and Experiment* 2005 (09) (2005) P09008.

- [11] C. H. Ding, X. He, H. Zha, M. Gu, H. D. Simon, A min-max cut algorithm for graph partitioning, in: Proceedings of the International Conference on Data Mining, 2001, pp. 107–114.
- [12] Y. Dourisboure, F. Geraci, M. Pellegrini, Extraction and classification of dense communities in the web, in: Proceedings of the 16th international conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 461–470.
- [13] M. Ester, H. Kriegel, S. Jörg, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the International Conference on Knowledge Discovery from Data, 1996, pp. 226–231.
- [14] T. Falkowski, A. Barth, M. Spiliopoulou, DENGGRAPH: a density-based community detection algorithm, in: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society, Washington, DC, USA, 2007, pp. 112–115.
- [15] S. Fortunato, Community detection in graphs, Physics Reports 486 (3-5) (2010) 75–174.
- [16] M. Girvan, M. E. Newman, Community structure in social and biological networks, in: Proceedings of the National Academy of Sciences, vol. 99, 2002, pp. 7821–7826.
- [17] D. Greene, D. Doyle, P. Cunningham, Tracking the evolution of communities in dynamic social networks, in: Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on, 2010, pp. 176–183.
- [18] M. S. Handcock, A. E. Rafter, J. M. Tantrum, Model-based clustering for social networks, Journal of the Royal Statistical Society A 170 (2007) 301–354.
- [19] J. Huang, H. Sun, J. Han, H. Deng, Y. Sun, Y. Liu, Shrink: a structural clustering algorithm for detecting hierarchical communities in networks, in: Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10, ACM, New York, NY, USA, 2010, pp. 219–228.
- [20] L. Hubert, P. Arabie, Comparing partitions, Journal of Classification 2 (1985) 193–218.
- [21] B. W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell System Technical Journal 49 (1970) 291–307.
- [22] M.-S. Kim, J. Han, Chronicle: A two-stage density-based clustering algorithm for dynamic networks, in: Proceedings of the 12th International Conference on Discovery Science, DS '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 152–167.
- [23] P. Kumar, L. Wang, J. Chauhan, K. Zhang, Discovery and visualization of hierarchical overlapping communities from bibliography information, in: Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 664–669.
- [24] J. M. Kumpula, M. Kivelä, K. Kaski, J. Saramäki, Sequential algorithm for fast clique percolation, Phys. Rev. E 78 (2008) 026109.

- [25] H. Y. Lam, A Learning Approach to Spam Detection Based on Social Networks, Hong Kong University of Science and Technology, 2007.
- [26] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, *Physical Review E* 80 (2009) 016118.
- [27] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, *New Journal of Physics* 11 (2009) 033015.
- [28] A. Lancichinetti, F. Radicchi, J. J. Ramasco, S. Fortunato, Finding statistically significant communities in networks, *PLoS ONE* 6 (5).
- [29] J. Leskovec, L. A. Adamic, B. A. Huberman, The dynamics of viral marketing, *ACM Trans. Web* 1 (1).
- [30] W. Li, Revealing network communities with a nonlinear programming method, *Inf. Sci.* 229 (2013) 18–28.
- [31] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, B. L. Tseng, Analyzing communities and their evolutions in dynamic social networks, *ACM Trans. Knowl. Discov. Data* 3 (2009) 8:1–8:31.
- [32] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, S. M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, *Behavioral Ecology and Sociobiology* 54 (2003) 396–405.
- [33] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, University of California Press, 1967, pp. 281–297.
- [34] J. B. MacQueen, Some methods for classification and analysis of multivariate observations, in: L. M. L. Cam, J. Neyman (eds.), *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, University of California Press, 1967, pp. 281–297.
- [35] A. McDaid, N. Hurley, Detecting highly overlapping communities with model-based overlapping seed expansion, in: *Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining, ASONAM '10*, IEEE Computer Society, Washington, DC, USA, 2010, pp. 112–119.
- [36] D. L. Nelson, L. C. McEvoy, T. A. Schreiber, *The University of South Florida word association, rhyme, and word fragment norms* (1998).
- [37] M. E. Newman, M. Girvan, Finding and evaluating community structure in networks, *Physical Review E* 69.
- [38] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, *Nature* 435 (7043) (2005) 814–818.
- [39] G. Palla, I. J. Farkas, P. Pollner, I. Deryni, T. Vicsek, Fundamental statistical features and self-similar properties of tagged networks, *New Journal of Physics* 10 (12) (2008) 123026.

- [40] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, *Proceedings of the National Academy of Sciences* 101 (9) (2004) 2658.
- [41] J. Reichardt, S. Bornholdt, Statistical mechanics of community detection, *PHYS.REV.E* 74 (2006) 016110.
- [42] M. Rosvall, C. Bergstrom, Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems, *PLOS ONE* 6 (2011) e18209.
- [43] M. Rosvall, C. T. Bergstrom, Maps of random walks on complex networks reveal community structure, *Proceedings of the National Academy of Sciences of the United States of America* 105 (4) (2008) 1118–1123.
- [44] J. P. Scott, *Social Network Analysis: A Handbook*, 2nd ed., Sage Publications Ltd, 2000.
- [45] H. Shen, X. Cheng, K. Cai, M.-B. Hu, Detect overlapping and hierarchical community structure in networks, *Physica A: Statistical Mechanics and its Applications* 388 (8) (2008) 1706.
- [46] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [47] J. Stehl, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. V. den Broeck, C. Rgis, B. Lina, P. Vanhems, High-resolution measurements of face-to-face contact patterns in a primary school, *CoRR* abs/1109.1015.
- [48] H. Sun, J. Huang, J. Han, H. Deng, P. Zhao, B. Feng, gskeletonclu: Density-based network clustering via structure-connected tree division or agglomeration, in: *Proceedings of the 2010 IEEE International Conference on Data Mining, ICDM '10*, IEEE Computer Society, Washington, DC, USA, 2010, pp. 481–490.
- [49] P. Sun, L. Gao, S. H. S., Identification of overlapping and non-overlapping community structure by fuzzy clustering in complex networks, *Inf. Sci.* 181 (6) (2011) 1060–1071.
- [50] P. G. Sun, L. Gao, Y. Yang, Maximizing modularity intensity for community partition and evolution, *Inf. Sci.* 236 (2013) 83–92.
- [51] B. Viswanath, A. Mislove, M. Cha, K. P. Gummadi, On the evolution of user interaction in facebook, in: *Proceedings of the 2nd ACM workshop on Online social networks*, ACM, 2009, pp. 37–42.
- [52] S. Wasserman, K. Faust, *Social Network Analysis: Methods and Applications*, Cambridge University Press, 1994.
- [53] F. Wei, C. Wang, L. Ma, A. Zhou, Detecting overlapping community structures in networks with global partition and local expansion., in: Y. Zhang, G. Yu, E. Bertino, G. Xu (eds.), *APWeb*, vol. 4976 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 43–55.
- [54] B. Wellman, Community: from neighborhood to network, *Communications of the ACM* 48 (10) (2005) 53–55.



- [55] S. White, P. Smyth, A spectral clustering approach to finding communities in graphs, in: Proceedings of the 5th SIAM International Conference on Data Mining, 2005, pp. 76–84.
- [56] C. Wilson, B. Boe, A. Sala, K. P. Puttaswami, B. Y. Zhao, User interactions in social networks and their implications, in: Proceedings of the 4th ACM European Conference on Computer Systems, ACM, New York, 2009, pp. 205–218.
- [57] J. Xie, S. Kelley, B. K. Szymanski, Overlapping community detection in networks: the state of the art and comparative study, CoRR abs/1110.5813.
- [58] J. Xie, B. K. Szymanski, Towards linear time overlapping community detection in social networks, in: Proceedings of the 16th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part II, PAKDD'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 25–36.
- [59] J. Xie, B. K. Szymanski, X. Liu, SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, in: Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops, ICDMW '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 344–349.
- [60] X. Xu, N. Yuruk, Z. Feng, T. A. J. Schweiger, SCAN: a structural clustering algorithm for networks., in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '07), ACM, 2007, pp. 824–833.
- [61] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33 (1977) 452–473.
- [62] O. R. Zaïane, J. Chen, R. Goebel, Dbconnect: Mining research community on dblp data, in: Proceedings of the Joint 9th WEBKDD and 1st SNA-KDD Workshop, 2007.
- [63] S. Zhang, R.-S. Wang, X.-S. Zhang, Identification of overlapping community structure in complex networks using fuzzy c-means clustering, Physica A: Statistical Mechanics and its Applications 374 (1) (2007) 483–490.

## APPENDIX-1

### Adjusted Random Index (ARI):

ARI [20] is a measure to compare two disjoint partitions ( $C_1$  and  $C_2$ ) of a set of  $n$  nodes of a network and is given by equation 16, where  $r_u(C_1, C_2)$  (the unadjusted rand index) is the fraction of pairs that belong to either same community or to different communities in both partitions  $C_1$  and  $C_2$  given by equation 17, and  $r_e(C_1, C_2)$  is the expected value of the same fraction in the null model given by equation 18.

$$ARI(C_1, C_2) = \frac{r_u(C_1, C_2) - r_e(C_1, C_2)}{1 - r_e(C_1, C_2)} \quad (16)$$

$$r_u(C_1, C_2) = \frac{|s(C_1) \cap s(C_2)| + |d(C_1) \cap d(C_2)|}{N} \quad (17)$$

$$r_e(C_1, C_2) = \frac{|s(C_1)||s(C_2)| + |d(C_1)||d(C_2)|}{N^2} \quad (18)$$

In equations 17 and 18,  $s(C)$  is the set of node pairs that belong to same community in  $C$ ,  $d(C)$  is the set of node pairs that belong to different communities in  $C$ , and  $N = n(n - 1)/2$  is the number of all possible pairs of  $n$  nodes of the network.

## Omega:

ARI measure has been generalized to an **Omega** measure for comparing overlapping partitions in [9] and given by equation 19.

$$\omega(C_1, C_2) = \frac{\omega_u(C_1, C_2) - \omega_e(C_1, C_2)}{1 - \omega_e(C_1, C_2)} \quad (19)$$

Unlike disjoint communities, overlapping communities can have a node pair which occurs in more than one community and thus the sum of equation 17 and the products for equation 17 have to run over all possible values till the maximum number  $m_{max} = \max(m(C_1), m(C_2))$  of communities in the two partitions. It means that for equation 19,  $\omega_u(C_1, C_2)$  is given by equation 20 and  $\omega_e(C_1, C_2)$  is given by equation 21.

$$\omega_u(C_1, C_2) = \frac{1}{N} \sum_{j=0}^{m_{max}} |t_j(C_1) \cap t_j(C_2)| \quad (20)$$

$$\omega_e(C_1, C_2) = \frac{1}{N^2} \sum_{j=0}^{m_{max}} |t_j(C_1) \cdot t_j(C_2)| \quad (21)$$

In equations 20 and 21,  $t_j(C)$  is the set of node pairs that occur  $j$  times together in a community of partition  $C$ .

## Normalized Mutual Information (NMI):

NMI [10] is another measure used to compare two disjoint partitions ( $C_1$  and  $C_2$ ) of a set of  $n$  nodes of a network, given by equation 22, where  $H(X)$  and  $H(Y)$  are the entropies of the random variables  $X$  and  $Y$  associated with the partitions  $C_1$  and  $C_2$ , respectively, and  $H(X, Y)$  is a joint entropy.

$$NMI(X : Y) = \frac{H(X) + H(Y) - H(X, Y)}{(H(X) + H(Y))/2} \quad (22)$$

Lancichinetti et al. [27] on the other hand defined a normalization of the *variation of information* as given in equation 23 which is interpreted as the average relative lack of information to infer  $X$  given  $Y$ , and vice versa.

$$V'_{norm}(X, Y) = \frac{1}{2} \left( \frac{H(X|Y)}{H(X)} + \frac{H(Y|X)}{H(Y)} \right) \quad (23)$$

## Overlapping-NMI (O-NMI):

Based on equation 23, Lancichinetti et al. defined a *normalized mutual information* measure for overlapping partitions as shown in equation 24.

$$O\text{-}NMI(X|Y) = 1 - \frac{1}{2} \left( \frac{H(X|Y)}{H(X)} + \frac{H(Y|X)}{H(Y)} \right) \quad (24)$$