

A Density-Based Approach to Detect Community Evolutionary Events in Online Social Networks

Muhammad Abulaish and Sajid Yousuf Bhat

Abstract With the advent of Web 2.0/3.0 supported social media, Online Social Networks (OSNs) have emerged as one of the popular communication tools to interact with similar interest groups around the globe. Due to increasing popularity of OSNs and exponential growth in the number of their users, a significant amount of research efforts has been diverted towards analyzing user-generated data available on these networks, and as a result various community mining techniques have been proposed by different research groups. But, most of the existing techniques consider the number of OSN users as a fixed set, which is not always true in a real scenario, rather the OSNs are dynamic in the sense that many users join/leave the network on a regular basis. Considering such dynamism, this chapter presents a density-based community mining method, *OCTracker*, for tracking overlapping community evolution in online social networks. The proposed approach adapts a preliminary community structure towards dynamic changes in social networks using a novel density-based approach for detecting overlapping community structures and automatically detects evolutionary events including *birth*, *growth*, *contraction*, *merge*, *split*, and *death* of communities with time. Unlike other density-based community detection methods, the proposed method does not require the neighborhood threshold parameter to be set by the users, rather it automatically determines the same for each node locally. Evaluation results on various datasets reveal that the proposed method is computationally efficient and naturally scales to large social networks.

Muhammad Abulaish

Department of Computer Science, Jamia Millia Islamia, New Delhi-25, India, e-mail: abulaish@ieee.org

Sajid Yousuf Bhat

Department of Computer Science, Jamia Millia Islamia, New Delhi-25, India, e-mail: s.yousuf.jmi@gmail.com

1 Introduction

With increasing popularity of Online Social Networks (OSNs) and their wide applications in different walk of life, community mining research has attracted researchers from various fields including data mining, web mining, and network science in recent past and the field is still rapidly evolving. As a result, various methods based on spectral clustering [7, 33], partitional clustering [22], modularity optimization [25], and latent space clustering [14] have been developed to identify users' communities in social networks. The fact that a person may have different diverse interests and consequently she may participate in more than one community has resulted in an increased attention towards detecting overlapping communities in social networks, and a solution based on k -clique percolation given by Palla et al. [27] is a step towards this end, followed by other density-based community detection methods, including `gSkeletonClu` [30], `CHRONICLE` [16], and `CMiner` [3] that are based on DBSCAN [9].

One of the important properties of the real-world social networks is that they tend to change dynamically as most often: i) new users join the network, ii) old users leave the network, and iii) users establish/break ties with other users. Consequently, all these evolutionary events result in *birth*, *growth*, *contraction*, *merge*, *split*, and *death* of communities with time. Although a number of community finding techniques have been proposed by different researchers, the dynamic nature of the real-world social networks (specifically, the online social networks like Facebook and Twitter) has been largely ignored in terms of community detection. In case of dynamic social networks, most of the studies either analyze a single snapshot of the network or an aggregation of all interactions over a possibly large time-window. But, such approaches may miss important tendencies of dynamic networks and in fact the possible causes of this dynamic behavior may be among the most important properties to observe [31]. Although, recent literature includes some approaches for analyzing communities and their temporal evolution in dynamic networks, a common weakness in these studies is that communities and their evolutions have been studied separately. As pointed out in [20], a more appropriate approach would be to analyze communities and their evolution in a unified framework, where community structure provides evidence about community evolution.

Considering the case of OSNs like Facebook and Twitter, community structures have mostly been analyzed using traditional community detection techniques over social networks representing explicit relations (friends, colleagues, etc.) of users. However, the observations made by Wilson et al. [34] and Chun et al. [5] on Facebook friendship and interaction data reveals that for most users, majority of their interactions occur only across a small subset of their social links, proving that only a subset of social links actually represents interactive relationships. Their findings suggest that social network-based systems should be based on the activity network, rather than on the social link network.

This paper presents the design of a density-based unified method, *OCTracker*, to identify overlapping communities and track their evolution in online social networks. The initial version of this work has been published as a short paper in

proceedings of the ASONAM'12 [4], and the major enhancement is the enhancement of the proposed methodology and the addition of more experimental results on different datasets. The proposed method detects dynamic overlapping community structures by automatically highlighting evolutionary events like *birth*, *growth*, *contraction*, *merge*, *split*, and *death* with time using a density-based approach. The novelty of the method lies in its overlapping community detection approach, which does not require the neighborhood threshold ε (mostly difficult to determine for density-based community detection methods) to be specified by the users manually. In addition, the proposed method is scalable to large social networks.

The rest of the chapter is organized as follows. Section 2 presents a brief review of the related works. Section 3 defines the similarity function and presents the density-based overlapping community detection approach. Section 4 describes the proposed approach for tracking evolution of overlapping communities in dynamic social networks. Section 6 presents the parameter estimation process, followed by a brief explanation of the overlapping community merging process in section 6. Section 7 presents the experimental setup and evaluation results. Finally, section 8 concludes the paper.

2 Related Work

Traditional community finding approaches are generally based on either graph partitioning methods [15] or partition-based clustering [2, 23], where the problem is to divide the nodes into k clusters by optimizing a given cost function. However, the main drawback of these methods lie in the requirement of the number of clusters and their sizes a priori. Hierarchical clustering is another well-known technique used in social network analysis [28, 32]. Starting from a partition in which each node is in its own community or all nodes are in the same community, one merges or splits clusters according to a topological measure of similarity between nodes. Other similar methods include methods based on the sociological notion of betweenness centrality [12] and methods based on modularity Q optimization [25].

Extending the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [9] to undirected and un-weighted graph structures, Xu et al. [35] 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 [16] is a two-stage extension of SCAN to detect the dynamic behavior of communities in dynamic networks. Similarly, considering only the weighted interaction graph of the online social networks, Falkowski et al. [10] extended the DBSCAN algorithm [9] to weighted interaction graph structures of online social networks. Some important features of density-based community detection methods include less computation, detection of outliers and natural scalability to large networks. However, the main drawback of traditional density-based community detection methods is that they require the global neighborhood threshold, ε , and the minimum cluster size, μ , to be specified

by the users. The methods are particularly sensitive to the parameter, (ϵ), which is difficult to determine. As an alternative, the method proposed in [30] reduces the number of possible values to consider for ϵ significantly by considering only the edge weights of a Core-Connected Maximal Spanning Tree (CCMST) of the underlying network.

The most popular method for identifying overlapping communities is the Clique Percolation Method (CPM) proposed by Palla et al. [27], which is based on the concept of k -clique, i.e., a complete subgraph of k nodes. The method relies on the observation that communities seem to consist of several small cliques that share many of their nodes with other cliques in the same community. In [24], the authors presented an overlapping community detection method `MOSES` by combining local optimization with Overlapping Stochastic Block Modeling (OSBM) [18] 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.

In order to find communities in dynamic social networks and to track their evolutions, various methods have been proposed recently. A typical dynamic community detection problem is formulated in [1, 31]. In these works, along a discrete timescale and at each time-step, social interactions of certain individuals of a network are observed and several subgraphs are formed. Based on these subgraphs, the true underlying communities and their developments over time are identified, so that most of the observed interactions can be explained by the inferred community structure. Similar approaches have been followed in [26, 13, 16]. However, as pointed out in [20], a common weakness in these approaches is that communities and their evolution are studied separately. It would be more appropriate to analyze communities and their evolution in a unified framework where community structure provides evidence about the community evolutions. Along this direction, [11] proposed a framework for studying community dynamics where a preliminary community structure adapts to dynamic changes in a social network. Our approach is similar to [11], but unlike it, our concern is on tracking the evolution of overlapping communities and we do not need an ageing function to remove old interactions from the network. Moreover, our method is applicable to directed/un-directed and weighted/un-weighted networks, whereas [11] applies only to un-directed and weighted networks. For un-weighted networks, the proposed method considers a unit weight for each edge in the network without altering the meaning or representation of the network.

3 Proposed Method

In this section we present the procedural detail of the proposed method to identify community evolution events. Along the lines of the `SCAN` [35], `DENGRAPH` [10], and other density-based community detection methods like `gSkeletonClu` [30] and `CHRONICLE` [16], the proposed method is based on `DBSCAN` [9]. As pointed out in section 2, the main drawback of traditional density-based community detec-

tion methods is that they require the global neighborhood threshold, ε , and the minimum cluster size, μ , to be specified by the users. On the other hand, the proposed method does not require the global neighborhood threshold parameter, ε , 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 using a much simpler approach from the underlying social network. Moreover, a local version of μ is also computed for each node automatically using a global percentage parameter η . The proposed method thus requires only a single tunable parameter η to be set by the users.

3.1 Distance Function and Parameter Estimation

This section presents a formal definition of a novel distance function and related concepts that are used in the proposed density-based overlapping community finding algorithm. The distance function defines distance between a pair of nodes by taking into consideration the average number of reciprocated interactions between the nodes and their commonly interacted nodes in the network. Considering the social network as a graph $G = (V, E)$, where V is the set of nodes representing users and $E \subseteq V \times V$ is the set of links between the users based on their interactions in the network, the distance function can be defined formally in the following paragraph. For simplicity, the symbols used throughout this paper and their interpretations are presented in table 1.

Table 1: Notations and their descriptions

Notation	Description
V	Set of nodes in the social network
E	Set of links in the social network
$I_{\vec{p}}$	Total number of out-going interactions of a node p
$I_{\vec{p}q}$	Number of interactions from node p to node q
$I_{\vec{p}q}^{\leftrightarrow}$	Reciprocated interactions of p and q : $\min(I_{\vec{p}q}, I_{\vec{q}p})$
$I_{\vec{p}}^{\leftrightarrow}$	Number of reciprocated interactions of a node p : $\sum_{q \in V_p} \min(I_{\vec{p}q}, I_{\vec{q}p})$
V_p	Set of nodes in the network with whom node p interacts
V_{pq}	Set of nodes with whom both nodes p and q interact: $V_p \cap V_q$

Definition 1 (Distance). For any two interacting nodes $p, q \in V$, the distance between them is represented as $\Delta(p, q)$ and defined as the minimum of the reciprocals of their mutual directed responses, normalized by their respective total count of out-going interactions in the interaction graph, as shown in equation 1.

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

In equation 1, $\delta(p, q)$ represents the response of node q to the interactions of node p , 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 2, where V_{pq} and $I_{\vec{pq}}$ have same interpretations as given in table 1.

$$\delta(p, q) = \begin{cases} \left(\frac{\sum_{s \in V_{pq}} (I_{\vec{ps}}) + I_{\vec{pq}}}{|V_{pq}| + 1} \right) & \text{if } I_{\vec{pq}} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Smaller values for $\Delta(p, q)$ represent higher response between the nodes p and q and thus represent more closeness between p and q , whereas higher values for $\Delta(p, q)$ translates to higher distance and thereby less closeness between the nodes p and q .

Definition 2 (Local-Neighborhood Threshold). For a node $p \in V$, the local neighborhood threshold is represented as ε_p and defined using equation 3 as the average per-receiver reciprocated interaction-score of p with all its neighbors (i.e., friends and non-friends with whom it interacts).

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

In equation 3, $\frac{I_{\vec{p}}}{|V_p|}$ represents the average number of reciprocated interactions between a node p and all other nodes in V to whom p sends interactions. The denominator $I_{\vec{p}}$ represents the total count of outgoing interactions of node p in the interaction graph and it has been used to normalize the value of ε_p to the range $[0, 1]$.

Definition 3 (Local ε -neighborhood). The local ε -neighborhood of a node $p \in V$ is represented by $N_{local}p$ and defined as the set of nodes to which p sends interactions such that the distance between p and each node in $N_{local}p$ is less than or equal to ε_p . Formally, the local ε_p -neighborhood of a node p can be given by equation 4.

$$N_{local}p = \{q : q \in V_p \wedge distance(p, q) \leq \varepsilon_p\} \quad (4)$$

For our proposed method, we define a local version of minimum-number-of-points for a node p , represented by μ_p , which is also computed automatically from the underlying social network. However, we need to specify a fraction η between $[0.0-1.0]$ to compute μ_p for a node p . For a node $p \in V$, the value of μ_p is taken as the fraction η of its interacted nodes in the network.

It should be noted that the fraction η , forms the only parameter for the proposed method to be set by the users. Moreover, besides determining the local minimum-number-of-points threshold, μ_p , for a node p , the value of η is also used to specify a distance constraint, which is specified as follows. The distance between two interacting nodes p and q can be measured by equation 1 only if the number of commonly

interacted nodes of p and q is greater than the number of nodes defined by the fraction η of the minimum of their individually interacted nodes minus one. Otherwise, the distance between them is taken as 1. Formally, the distance constraint can be specified using equation 5.

$$distance(p, q) = \begin{cases} \Delta(p, q) & \text{if } |V_{pq}| > (\eta \times \min(|V_p|, |V_q|)) - 1 \\ 1 & \text{otherwise} \end{cases} \quad (5)$$

Definition 4 (Core node). A node $p \in V$ having non-zero reciprocated interactions is defined to be a core node with respect to a global percentage constant η , if its local ε_p -neighborhood contains at least μ_p (local minimum-number-of-points threshold for p) of its interacted nodes.

The proposed method identifies core nodes and uses them to grow communities in a recursive manner using the following definitions. It should be noted that all the definitions used in the proposed method are significantly different from the definitions used in traditional density-based community detection methods in terms of the overall concept used to define a community.

Definition 5 (Direct density-reachability). A node q is direct density-reachable from a node p with respect to η if p is a core node and q belongs to the local ε_p -neighborhood of p .

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 6 (Mutual cores). Two nodes p and q are called mutual cores if both p and q are core nodes, and p belongs to the local ε_q -neighborhood of q , and q belongs to the local ε_p -neighborhood of p . In other words, two nodes p and q are mutual cores if they are direct density-reachable from each other.

Definition 7 (Density reachability). A node q is density-reachable from a node p with respect to η , 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 ranging from $1, 2, \dots, n-2$, and v_n is direct density-reachable from v_{n-1} .

Definition 8 (Density connectivity). A node q is density-connected to a node p with respect to η , if there exists a node v such that both p and q are density reachable from v .

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

Definition 9 (Density-connected community). A non-empty subset $C \subseteq V$ is called a density-connected community with respect to η , if all the vertices in C are density-connected with each other and C is maximal with respect to density reachability.

3.2 *Overlapping Community Detection*

In order to identify overlapping communities in social network data, initially all nodes of the network are un-labeled and un-visited. For a given global percentage threshold, η , the 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. 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 the local ϵ_p threshold for p using equation 3. If the ϵ_p threshold for p is greater than zero, then it uses the distance function of equation 1 and distance constraint to determine the local ϵ_p -neighborhood of p , i.e., $N_{local}p$. 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 $N_{local}p$ is also appended with the same. We use the term appended as the nodes belonging to $N_{local}p$ 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 for identifying a density-connected community of p .

1. All un-visited mutual-core nodes of node p in $N_{local}p$ 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 in order 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 $N_{local}p$ forms a community with p being its only primary core-node.
2. If a core-node q in $N_{local}p$ 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 un-altered.
3. Non-core nodes in $N_{local}p$ are marked as visited and they are appended with the current community label. Such nodes form boundary nodes for the community of p and are thus not pushed into the stack as they cannot be used to grow a community.

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 . Similarly, 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 the overlapping community structure in the social network. At the end of the process, 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 any node or enough number of nodes in the social network.

4 Community Evolutionary Events Tracking

It should be noted that unlike [11], we do not need an ageing function to remove old interactions and we also argue that it is difficult to decide upon a selection criteria to do so. As our approach involves local average interactions of nodes for the clustering process, addition of new interactions results in new averages for the involved nodes and directly effects their neighborhoods and roles for clustering. A social network and its resulting community structure can evolve due to various events triggered by the social network individuals. These events may include:

1. Addition of new weighted interaction links and/or nodes
2. Increase in the interaction weights of existing links
3. Removal of existing nodes

In order to track the evolution of communities in dynamic social networks like OSNs, the proposed method first detects a preliminary community structure from an initial state of the network using the method discussed in section 3.2. Then for each node involved in a change in the network, i.e., the events mentioned earlier, various transitions can occur. They can be handled by either considering a live stream of changes as the network evolves (an online evolutionary adaption of the community structure), or the set of changes observed in a specific time-window (an offline evolutionary adaption of the community structure). In either case, the new edges and/or nodes are added to the network or nodes are removed from the network, and each node involved in a change and its direct-neighbors (nodes with which they have an edge) in the network are marked as un-visited. The reason to consider the direct-neighbors also is that in our proposed method the local ϵ_p -neighborhood of a node is also dependent on the interaction behavior of its direct-neighbor(s) in a network. So if a node p interacts with some other node q , besides re-determining the local ϵ_p -neighborhoods of p and q we also need to re-determine the local ϵ_p -neighborhoods of all the immediate neighbors of p and q respectively to detect the induced change by the nodes p and q . Thereafter, each remaining un-visited node is re-checked for a core-node property by re-calculating its local $\epsilon_{(p)}$ -neighborhood. Various events or transitions used by proposed method to model the evolution of communities are presented in the following sub-sections.

4.1 A Non-Core Node Becomes a Core

In this case, either an existing non-core node or a newly added node in the network becomes a core node. In order to track a possible evolutionary event, the following conditions are checked.

For the new core node p , if there exist core nodes in the local $\varepsilon_{(p)}$ -neighborhood with which the node p has mutual-core relations and which already belong to different communities, then p causes the primary communities of these core nodes to *merge* into a single community. Consequently, in this case, p causes the MCMSs of different communities to join and form a single MCMS for the new merged community. The merged community also forms the primary community of the new core node p and nodes in its local neighborhood are also added to the merged community.

If the new core node p has mutual-core relations with nodes that have the same primary community C , then p also forms a primary core of community C by appending this community label to itself and to its local neighborhood. This simply results in the *expansion* of community C .

Finally, if the new core node p has no mutual-core relations, then p forms a new community and appends the new community label to its local neighborhood and itself. This causes the *birth* of a new community with p being its only primary core.

4.2 A Core Node Becomes a Non-Core

In this case, an existing core node no longer remains a core node due to some change in the network. This triggers either a *split* or a *shrink* event in the evolution of a community as follows.

Let p be a primary core node of a community C at a previous stage, and p cease to exist as a core node due to a new dynamic change in the network. Let S be the set of primary cores of the community C which had mutual-core relations with p before the change in the network. We mark the nodes in S as un-visited. For any core node $q \in S$, let T be a simple BFS (Breadth First Search) traversal of nodes starting from q , visiting nodes in the local neighborhoods of the core nodes and branching at mutual-core relations wherein each newly visited node is labeled as visited. If T includes all the core nodes in S , then p is simply removed from being a primary core of community C . Moreover, if p and/or any other node that belonged to the earlier local neighborhood of p are not in the traversal T , then they are removed with the community label of C , causing C to *shrink*.

However, If T does not include all the core nodes in S , then T forms a new community, i.e., the original community C *split* as p with loosed core-node property causes a cut in the MCMS of C . The community label C of the nodes in T (which now represents a split part of community C) are replaced with a new community label. The traversals are repeated for each remaining un-visited core nodes in S until no further split of community C is possible, i.e., no node in S remains un-visited after a traversal. In the last traversal, if a node s is visited which does not have the

community label of C (i.e., it was removed as s belonged to a previous traversal that split the community C), then the community label of C is re-appended to it resulting in an overlapping node. At the end, the node p and/or any node that belonged to its previous local neighborhood may be labeled with community label C , but do not belong to the last traversal. In this case, the community label C for these nodes is removed, causing community C to further *shrink*.

It is also worth to note that in case a lost core node p was the only primary core node of a community C , then p with loosed core-node property causes the *death* of community C as no representative primary core node for community C remains.

4.3 A Core Node Gains/Looses Nodes but Remains as Core

Due to dynamic nature of social networks, changes in them may cause a core node to gain or loose nodes or both but still hold the core node property. In this case, the addition or removal of nodes are handled as follows.

If the local ϵ_p -neighborhood of a core node p gains a set of nodes S that do not have *mutual-core* relation with p , then the *primary-community* label of p is simply appended to each node $q \in S$. However, if the added nodes have *mutual-core* relation with p , then they are handled in the same way as the *mutual-cores* of a newly formed core node are handled (section 4.1). This can either cause the *expansion* of a community or *merge* of multiple communities. It is obvious that if all the *mutual-cores* of p in its neighborhood including p have the same *primary-community*, then only the neighborhood of p is updated resulting in *expansion* of a community.

Consider the case when the local ϵ_p -neighborhood of a core node p with a *primary-community* C , looses a set of nodes L that were earlier in its ϵ_p -neighborhood. If the nodes in L do not have *mutual-core* relation with p , and they are not direct density-reachable from any other *primary-core* of the community C , then the community label of community C is removed from the lost nodes resulting in the *shrinkage* of community C . However, if a core node p looses a set of nodes S that had *mutual-core* relation with it, then such nodes are handled in the same way when the *mutual-core* of a core node no longer remains a core node (section 4.2). But, in this case the core node p in question is not excluded from the set of nodes S . This could possibly lead to either *split* or no change to the community C .

Most of the community dynamics can be tracked by considering only the three previously mentioned transitions or events and can be used to model the community-centric evolutionary events easily.

5 Parameter (η) Value Estimation

The proposed method requires only a single parameter, η , to be set by the users for detecting overlapping community structures in a social network. The value of

η basically defines the size and density of the overlapping communities to be detected. Smaller values of η yield larger and less-dense communities, whereas larger values yield smaller and more-dense communities. This implies that the parameter, η , can be tuned to detect overlapping community structures at different levels of granularity, naturally forming a hierarchical representation of the identified overlapping communities. In order to find a good approximation for η , the proposed method considers a minimum (η_{min}) and a maximum (η_{max}) values for η , and the community structures are identified from η_{min} to η_{max} at some regular step until the modularity score [25] of the community structure for the current step is no longer better (or same) than the previous step. In this way, the proposed method takes the value of η between η_{min} and η_{max} as the one where the modularity score of the identified community structure is highest. To define such a domain of η for an underlying network, the proposed method considers the *topological-overlap* (equation 6) between a pair (i, j) of reciprocating nodes¹.

$$\sigma_{Overlap} = \frac{|N_i \cap N_j|}{\min(|N_i|, |N_j|)}, \quad (6)$$

In equation 6, N_i and N_j represents the sets of nodes to which nodes i and j have out-links, respectively. The *mean* and *standard_deviation* of the *topological-overlap* are taken over all reciprocating pairs of nodes in the underlying network (rounded-up to two decimal places), and the value of *step* is taken as the *standard_deviation*/2 (rounded-up to two decimal places). The η_{min} value is determined as follows. If *mean* + *standard_deviation* is less than or equal to 0.5, then $\eta_{min} = \text{mean} + \text{step}$, otherwise $\eta_{min} = \text{mean}$ (truncated to one decimal place). The η_{max} value is taken as $\eta_{max} = \eta_{min} + \text{standard_deviation}$.

The above procedure is used to determine a good approximation of η for every consecutive state of a dynamic network. It is possible that the η_{min} value for a network state at time $t + 1$ is less than the η value decided for a previous network state at time t . In this case, all the nodes in the network at time $t + 1$ are marked as un-visited and the changes to the local ϵ -neighborhoods are determined for each node.

6 Overlapping Communities and Post-Merge

As mentioned earlier, the proposed community detection method identifies overlapping community structures in a social network. It does so by allowing a node q to belong to the ϵ_p -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 com-

¹ For a directed network two nodes are said to be reciprocating if each has an out-going edge towards the other, whereas for un-directed networks each edge is considered to represent a bi-directional reciprocal edge

munities overlap. It is often possible that two communities overlap in such a way that majority of nodes 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 in [8]. For the proposed method such a merging of highly overlapping communities is performed as follows. After a community structure is determined from some state of the underlying network at a particular value of η , the proposed method can merge two overlapping communities if the number of nodes, involved in the overlap between them, for the smaller community is more than or equal to the fraction η_{max} of its candidate nodes. In this work, the process of merging highly overlapping communities identified during any state of an underlying network is termed as *post-merge*.

7 Experimental Results

This section presents the experimental results of the proposed method on some benchmark datasets. We compare the results obtained through proposed method with four other state-of-the-art community detection methods that include MOSES [24], DENGGRAPH [10], gSkeletonClu [30], and CFinder [27]. The evaluation is performed based on two scoring measures which include *omega index* [6] and *normalized mutual information*(NMI) [17]. Both Omega and NMI are generalized scoring measures used for evaluating both overlapping and non-overlapping community structures.

gSkeletonClu and MOSES are parameter free methods and do not require an input. On the other hand, CFinder requires an input parameter k to define the clique size, which has been set to $k = 4$ in our experiment as the method generates best results for this clique size. For DENGGRAPH, the input parameters ϵ and μ have been varied to generate the best possible results. All the experiments were performed on an Intel i3 based computer with 4GB memory.

7.1 Results on Static Networks

For this experiment, we have used four well-known real-world benchmarks to evaluate the performance of the proposed method and compared it with other state-of-the-art. For all four real-world network datasets, the ground truth community structures are known and are used to calculate the performance scores. Figure 1 gives the comparison of the proposed method with other state-of-the-art methods on the benchmark datasets.

Figure 1a compares the result scores of the proposed method at $\eta = 62\%$ on the un-directed and weighted Zachary’s Karate club network [36] with other methods. The proposed method identifies a total of three overlapping communities out of

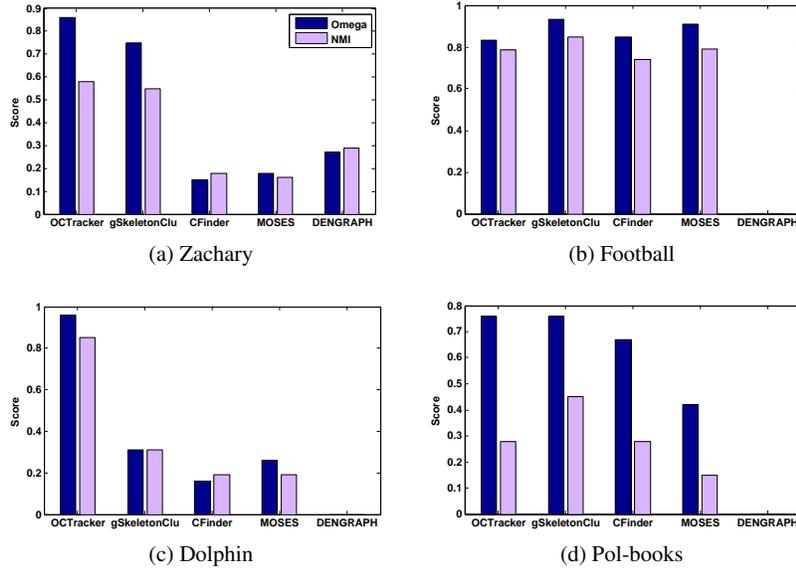


Fig. 1: Experimental results on real-world static datasets.

which two almost perfectly match the ground truth. The third community consists of only three nodes out of which one is involved in an overlap with other community resulting in only two misclassified nodes. It can be seen that the community structure identified by the proposed method scores better than all the other methods in question.

Figure 1b gives the comparison of the methods on a 2000 season NCAA College football network (un-directed and un-weighted) [12], which consists of 115 college football teams, divided into eleven conferences and five independent teams that do not belong to any conference. The proposed method at $\eta = 50\%$ exactly identifies eleven communities from the network that almost perfectly match the ground truth. It also identifies five independent teams that do not belong to any conference as outliers. However, it additionally marks three other nodes as outliers and one of the nodes is assigned to two conferences. Figure 1b concludes that almost all methods in question perform well and that the proposed method is comparable to the state-of-the-art methods.

Figure 1c compares the methods on an un-directed and un-weighted social network of frequent associations among 62 Dolphins in a community living off Doubtful Sound, New Zealand that has been compiled by Lusseau et al. [21]. The results obtained by the proposed method are at $\eta = 50\%$. It is clear from figure 1c that the proposed method performs marginally better than all other methods in question on the Dolphin network.

Figure 1d provides the comparison of the performance scores of the OCTracker with other methods on the US political books network (un-directed and un-weighted). This network is a dataset of books about US politics compiled by Valdis Krebs <http://www.orgnet.com/> wherein the nodes represent books about US politics sold online by Amazon and the edges represent frequent co-purchasing of books by the same buyers. Mark Newman <http://www-personal.umich.edu/~mejn/netdata/> clustered the nodes of this network into 'liberal', 'neutral' and 'conservative' based on the description and reviews of books posted on Amazon. The network consists of 105 nodes (books) and 441 edges (co-purchases). The proposed method identifies a total of five communities at $\eta = 62\%$ with four overlapping nodes and two outliers. Two of the identified communities closely match to the actual 'liberal' and 'conservative' categories. However, the 'neutral' category is difficult to identify and is scattered into three communities by the proposed method along with a few nodes from the 'liberal' and 'conservative' categories. Figure 1d shows that the proposed method also performs reasonably well on the political books network dataset. It is notable from figure 1 that DENGGRAPH [10] is not able to identify the community structure in un-weighted networks.

7.2 Results on Dynamic Networks

This section presents experimental results on two dynamic network datasets. The first dataset comprises two weighted networks of face-to-face proximity between 242 individuals representing students and teachers in a primary school over a period of two days [29]. The two networks correspond to two days of study wherein a daily contact network is provided. The nodes in this network represent students and teachers, and edges correspond to the interactions between them. The weight of an edge represents the number of times two nodes have interacted during the day. The students actually belong to ten different classes which can represent the ground truth communities. The teachers do not specifically belong to any class and interact with any student community. Our aim is to track the community-centric evolutionary events that possibly occur during the two days of interactions between the individuals. We also aim to see how well can the actual communities in the network, at various stages, be detected by the community detection methods.

Figure 2a shows the comparison of performance scores (Omega and NMI) for the various methods on the interaction network of the individuals after day-1. The scores are computed against the known ground truth for day-1. As can be seen from the results for day-1, the proposed method performs better than all other methods in question.

In order to detect the evolutionary events, the set of community structures detected by the proposed method for day-1 forms its initial community state. This initial community structure is now adapted to changes in the network, i.e., by adding interactions for day-2 to the underlying network which could also include adding new nodes, as discussed in section 4. Figure 3 shows the dynamic changes that

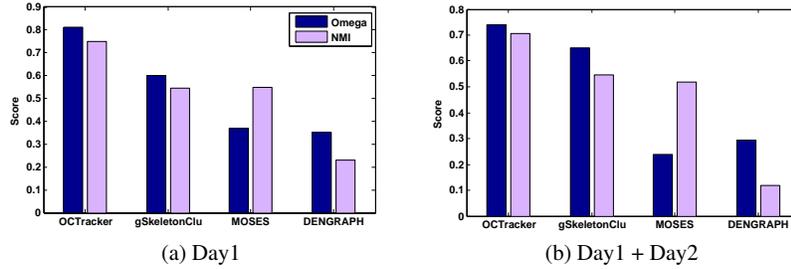


Fig. 2: Experimental results on a primary school dynamic network considering the network (a) after day-1, and (b) after day-2 (i.e., merged day-1 and day-2 network).

occur in the community structure of the primary school interaction network over two days as tracked by the proposed method. Initially on day-1 network, the proposed method detects 9 communities labeled as $A - H$, of which community C overlaps with D and E overlaps with I . The interactions for day-2 are merged with the underlying day-1 network which leads to addition of some new nodes and edges, and increases the weights of some already existing edges. Thereafter, OCTracker scans the changes in the network as discussed in section 4 and tracks the resulting community-centric changes in the initial community structure. As shown in figure 3, almost all the initial communities gain nodes resulting in their expansion. Two important evolutionary events are detected by the proposed method after the second day of interactions. Firstly, the two overlapping communities C and D merge to form a single community labeled as $C + D$. Secondly, community G splits into two overlapping communities labeled as G_1 and G_2 . Moreover, after the second day of interactions, many communities² begin to overlap with each other which are represented by overlapping circles in figure 3.

Figure 2b shows the comparison of performance scores (Omega and NMI) for the various methods on the interaction network of the individuals after day-2, i.e., the network represented by merging the interactions and nodes for both day-1 and day-2. The scores are computed against the known ground truth for both day-1 and day-2 data. As can be seen from the results, the proposed method again performs better than all other methods in question for the complete primary school interaction network over two days. To generate the results for the proposed method on the primary school network dataset, the input parameter η is set to 65%. Surprisingly, CFinder could not generate any results for the primary school network data due to its higher space complexity.

The second dataset [19] is a dynamic directed-network of about 8000 users from the English Wikipedia that voted for and against each other in admin elections from year 2004 to 2008. Nodes represent individual users, and directed-edges represent

² Figure 3 does not depict the actual size of the detected communities or the amount of overlap between communities.

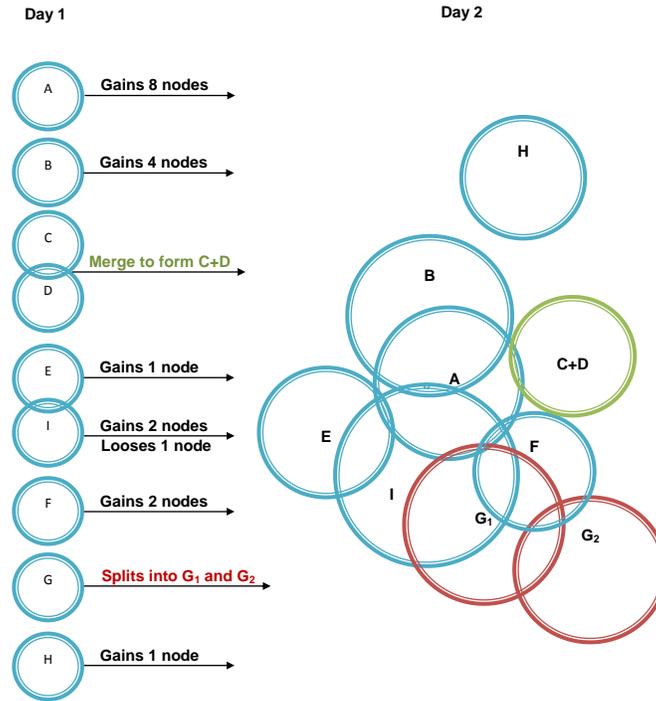


Fig. 3: Community evolution tracking in a primary school dynamic network.

votes. Edges are positive (“for” vote) and negative (“against” vote) represented by edge weights of 1 and -1 , respectively. For this paper, the dataset is divided into five subnetworks based on the year of voting. Starting with the network of year 2004, the proposed method identifies the preliminary community structures. Then for each subsequent year, it adds the respective subnetwork to the current state of the network and identifies the changes induced to the existing community structures for the new state of the network. The proposed method finds highly-overlapping communities from each state (cumulative network from the start to some later year) of the voting network. Some of the evolutionary transitions (birth, split, and merge) for some of the communities across any two consecutive states (years) of the voting network identified by the proposed method (without post-merge) is shown in figure 4. Based on these results, we conclude that the proposed method can identify the evolutionary transitions (birth, growth, contraction, merge, split, and death) of communities across a time-varying network even if the changes involve only the addition of new edges and/or nodes. It means that the proposed method does not necessarily require an ageing function to remove old links.

As mentioned earlier, the partial results on the Wikipedia election network shown in figure 4 are generated by the proposed method without performing the post-merge process. On applying post-merge to the community structure identified for

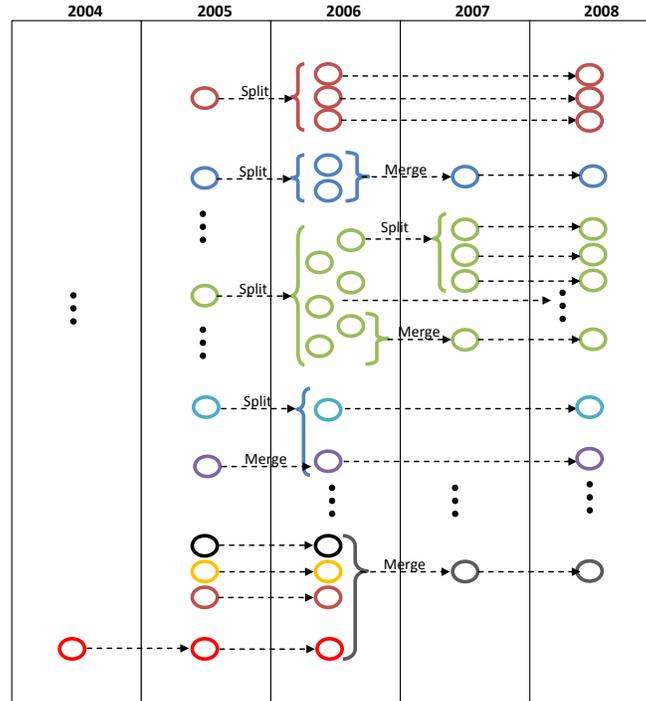


Fig. 4: Community tracking on the signed and directed Wikipedia election network [19].

each state of the network, the number of communities for each state are reduced as many highly overlapping communities are merged to represent a single community. The analysis of the community evolution trend, using post-merge with the proposed method, reveals that at every new state new nodes tend to join existing larger communities (and cause their growth) or form completely new communities instead of involving in merge or split.

8 Conclusion

This chapter has presented a novel density-based approach to track the evolution of overlapping community structures in online social networks. The novelty of the proposed method lies in the approach for allowing the communities to overlap, and its distance function which is defined as a function of the average interactions between a node and its neighborhood. In addition, unlike other density-based methods for which the neighborhood threshold is to be set by the users, which is generally dif-

difficult to determine, the proposed method computes a local neighborhood threshold for each node from the underlying network. The preliminary experimental results on both static and dynamic networks show that the proposed method is comparable to the state-of-the-art methods and can effectively track the evolutionary events in dynamic networks. The method is naturally scalable to large social networks.

References

1. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: membership, growth, and evolution. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06, pp. 44–54. ACM, New York, NY, USA (2006)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA (1981)
3. Bhat, S.Y., Abulaish, M.: A density-based approach for mining overlapping communities from social network interactions. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics (WIMS'12), pp. 9:1–9:7. ACM (2012)
4. Bhat, S.Y., Abulaish, M.: Octracker: A density-based framework for tracking the evolution of overlapping communities in osns. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'12). IEEE Computer Society (2012)
5. Chun, H., Kwak, H., Eom, Y., Ahn, Y., Moon, S., Jeong, H.: Comparison of online social relations in volume vs interaction: a case study of cyworld. In: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement, vol. 5247, pp. 57–70 (2008)
6. Collins, L.M., Dent, C.W.: Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research* **23**(2), 231–242 (1988)
7. Ding, C.H., He, X., Zha, H., Gu, M., Simon, H.D.: A min-max cut algorithm for graph partitioning. In: Proceedings of the International Conference on Data Mining, pp. 107–114 (2001)
8. Dourisboure, Y., Geraci, F., Pellegrini, M.: Extraction and classification of dense communities in the web. In: Proceedings of the 16th international conference on World Wide Web, WWW '07, pp. 461–470. ACM, New York, NY, USA (2007)
9. Ester, M., Kriegel, H., Jörg, S., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the International Conference on Knowledge Discovery from Data, pp. 226–231 (1996)
10. Falkowski, T., Barth, A., Spiliopoulou, M.: DENGGRAPH: a density-based community detection algorithm. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, pp. 112–115. IEEE Computer Society, Washington, DC, USA (2007)
11. Falkowski, T., Barth, A., Spiliopoulou, M.: Studying community dynamics with an incremental graph mining algorithm. In: Proceedings of the 14th Americas Conference on Information Systems (AMCIS) (2008)
12. Girvan, M., Newman, M.E.: Community structure in social and biological networks. In: Proceedings of the National Academy of Sciences, vol. 99, pp. 7821–7826 (2002)
13. Greene, D., Doyle, D., Cunningham, P.: Tracking the evolution of communities in dynamic social networks. In: Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining, ASONAM '10, pp. 176–183. IEEE Computer Society, Washington, DC, USA (2010)
14. Handcock, M.S., Rafter, A.E., Tantrum, J.M.: Model-based clustering for social networks. *Journal of the Royal Statistical Society A* **170**, 301–354 (2007)
15. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal* **49**, 291–307 (1970)

16. Kim, M.S., Han, J.: Chronicle: A two-stage density-based clustering algorithm for dynamic networks. In: Proceedings of the 12th International Conference on Discovery Science, DS '09, pp. 152–167. Springer-Verlag, Berlin, Heidelberg (2009)
17. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* **11**, 033,015 (2009)
18. Latouche, P., Birmel, E., Ambroise, C.: Overlapping stochastic block models. *Bernoulli in press*(25), 1–26 (2009)
19. Leskovec, J., Huttenlocher, D., Kleinberg, J.: Predicting positive and negative links in online social networks. In: Proceedings of the 19th international conference on World wide web, WWW '10, pp. 641–650. ACM, New York, NY, USA (2010). DOI 10.1145/1772690.1772756. URL <http://doi.acm.org/10.1145/1772690.1772756>
20. Lin, Y.R., Chi, Y., Zhu, S., Sundaram, H., Tseng, B.L.: Analyzing communities and their evolutions in dynamic social networks. *ACM Trans. Knowl. Discov. Data* **3**, 8:1–8:31 (2009)
21. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology* **54**, 396–405 (2003)
22. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press (1967)
23. MacQueen, J.B.: 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, pp. 281–297. University of California Press (1967)
24. McDaid, A., Hurley, N.: 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, pp. 112–119. IEEE Computer Society, Washington, DC, USA (2010)
25. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* **69** (2004)
26. Palla, G., Iászló Barabási, A., Vicsek, T., Hungary, B.: Quantifying social group evolution. *Nature* **446**, 664–667 (2007)
27. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**(7043), 814–818 (2005)
28. Scott, J.P.: *Social Network Analysis: A Handbook*, 2 edn. Sage Publications Ltd. (2000)
29. Stehl, J., Voirin, N., Barrat, A., Cattuto, C., Isella, L., Pinton, J.F., Quaggiotto, M., den Broeck, W.V., Rgis, C., Lina, B., Vanhems, P.: High-resolution measurements of face-to-face contact patterns in a primary school. *CoRR* **abs/1109.1015** (2011)
30. Sun, H., Huang, J., Han, J., Deng, H., Zhao, P., Feng, B.: 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, pp. 481–490. IEEE Computer Society, Washington, DC, USA (2010)
31. Tantipathananandh, C., Berger-Wolf, T., Kempe, D.: A framework for community identification in dynamic social networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '07, pp. 717–726. ACM, New York, NY, USA (2007)
32. Wasserman, S., Faust, K.: *Social Network Analysis: Methods and Applications*. Cambridge University Press (1994)
33. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: Proceedings of the 5th SIAM International Conference on Data Mining, pp. 76–84 (2005)
34. Wilson, C., Boe, B., Sala, A., Puttaswami, K.P., Zhao, B.Y.: User interactions in social networks and their implications. In: Proceedings of the 4th ACM European Conference on Computer Systems, pp. 205–218. ACM, New York (2009)
35. Xu, X., Yuruk, N., Feng, Z., Schweiger, T.A.J.: SCAN: a structural clustering algorithm for networks. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '07), pp. 824–833. ACM (2007)

36. Zachary, W.W.: An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **33**, 452–473 (1977)