Journal of Computational Science 25 (2018) 122–133

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

A cascade information diffusion based label propagation algorithm for community detection in dynamic social networks

Mohammad Sattari, Kamran Zamanifar*

Department of Computer Engineering, University of Isfahan, Isfahan, Iran

ARTICLE INFO

Article history:
Received 14 March 2017
Received in revised form 1 November 2017
Accepted 28 January 2018
Available online 15 February 2018

Keywords:
Label propagation approach
Community detection
Dynamic social network
Cascade information diffusion

ABSTRACT

One of the most important topics in social network analysis is community detection in dynamic social networks. A variety of approaches exists for detecting communities in dynamic social networks, among which the label propagation algorithm (LPA) is the well-known approach. This approach has made remarkable performance, but still has several problems. One of the difficulties of this approach is the new nodes added to the social network graph in the current snapshot has a very slight chance of creating new communities. In fact, these nodes fall under the influence of existing communities. This drawback decreases the accuracy of community detection in dynamic social networks. We propose a new method based on label propagation approach and the cascade information diffusion model in order to solve this difficulty. Here, the newly proposed method, Speaker Listener Propagation Algorithm Dynamic (SLPAD), Dominant Label Propagation Algorithm Evolutionary (DLPAE) and Intrinsic Longitudinal Community Detection (ILCD) on real and synthetic networks are implemented. The findings indicate that the modularity and Normalized Mutual Information (NMI) and also $F1_{AVG}$ of this proposed method is considerably higher than the earlier available methods in most datasets. Therefore, it can be concluded that the proposed method improves the accuracy of community detection in comparison with other available methods.

1. Introduction

In recent years, research on community detection has become a hot topic in social networks [1]. A community is a set of densely connected nodes that have weaker links with the rest of network [2]. Detection communities can be associated with clustering nodes based on the network topology while no information on the size and structure of communities is available [3]. Detecting communities can be both static and dynamic. In static mode, one snapshot is considered. In dynamic mode, more than one snapshot is considered. In fact, the dynamic mode is matched with more realistic results than static mode.

There are many approaches for detecting communities in social networks, among which label propagation approach (LPA) is a simple and time-efficient approach. In LPA, each node is a community with only one node [4]. Several methods based on LPA are proposed in order to detect communities in the static or dynamic networks such as SLPA [5], COPRA [6] and DLPAE [7]. These methods are fast and simple, while over-propagating the labels lead to the formation of monster communities. Formation of this kinds of communities decreases the accuracy of community detection based upon label propagation, especially in dynamic social networks.

For alleviating the limitations of LPA, this paper proposes a method based on the LPA approach and an information cascade model. We used this model based upon the observation that it often has been much more efficient within communities than among communities [8]. We apply the CID (Cascade Information Diffusion) model [9], which is one of the famous information diffusion models. In the CID model, each node has two states. The first state is S_0-state consisting of uninformed nodes, which tend to receive information. The second state is S_1-state consisting of the nodes, which try to cascade information diffusion to S_0-state neighbours. The proposed method has two main stages. The first principal stage consists of applying the CID model, and the second principal stage is LPA affected by the previous stage. We called this proposed method is "CIDLPA", where CID stands for cascade information diffusion and LPA stands for Label Propagation Algorithm.

The main contribution to this paper is threefold:

* Corresponding author.
E-mail addresses: mohammadsattari9220@eng.ui.ac.ir (M. Sattari), zamanifar@eng.ui.ac.ir (K. Zamanifar).

https://doi.org/10.1016/j.jocs.2018.01.004
1877-7503/

- It adopts the CID model on the label propagation algorithm in order to improve the accuracy of community detection in dynamic social networks.
- The proposed method can detect overlapping communities in a dynamic social network.
- Improved label propagation approach and accuracy.

The structure of this paper is organized as follows: Section 2 is a literature review. Section 3 describes notations and the concept of information diffusion. Section 4 presents the proposed method and the algorithm description. Moreover, the time complexity of this method is provided in this section. In Section 5, experimental results are discussed. Finally, Section 6 concludes the paper.

2. Literature review

2.1. Community detection

There are several community detection approaches in static networks such as graph partitioning [10], density-based clustering [11], modularity-based [12] and label propagation algorithm (LPA) [4]. The LPA is a simple and highly efficient approach which discovers communities by exchanging labels [4–6]. Raghavan et al. proposed LPA, which detects communities in social networks with linear time complexity [4]. However, this algorithm can only handle disjoint community detection. Gregory [6] improved the LPA by revealing overlapping communities, but this method leads to inappropriate results when there are too many mixed or overlapping communities. Xie et al. proposed the Speaker Listener Propagation Algorithm (SLPA), which detects overlapping communities based on LPA [5]. In SLPA, each node can have more than one label, where each label corresponds to one community. This method can execute in a parallel manner [13], but it has poor robustness in dense networks with unclear community structure. To solve this problem, Sun et al. presented a new method, which changes the update sequence of LPA approach [14]. The method presents a novel measure in order to compute the centrality of nodes. Then, it detects communities based on the sequence containing nodes are sorted ascending on centrality. Deng et al. proposed another method in order to solve the poor robustness problem of LPA in dense networks [8]. The method first applied LPA and then sorted the kernel node of each community on its importance.

A large number of community detection methods were applied in static network, even though most real-world social networks inherently are dynamic [15]. Therefore, the methods in static mode are extended for community detection in dynamic social networks. Lin et al. provided a framework for the analysing evaluation of disjoint communities in real networks, but this framework needs prior information such as the number of communities [16]. Nguyen et al. proposed Quick Community Adaptation (QCA) algorithm, which traced the community structure evolution and applied various strategies based on the added or removed nodes, and edges[15]. However, this method produces small communities, which decrease the accuracy of community detection in dynamic networks. Xie et al. provided LabelRankT, which mine communities by exchanging label distribution vectors [17].

All of the aforementioned dynamic methods try to find disjoint communities, but communities overlap in many real-world social networks [18]. Cazabet et al. [18] presented Intrinsic Longitudinal Community Detection (ILCD), which tries to find small communities growing gradually with the passage of time. This method could find overlapping communities, but it did not consider the contract and death of communities. Lancichinetti et al. presented Order Statistics Local Optimization Method (OSLOM), which considers the edge direction and weight to detect communities [19]. However, this method produces many singleton communities, which decrease the accuracy of community detection in dynamic networks.

SLPAD is an overlapping community detection method, which is based on label propagation approach [20]. It is a dynamic version of the SLPA method proposed in [5]. Although this method is fast, it tends to create monster communities. Dominant Label Propagation Algorithm Evolutionary (DLPAE) [7] is a dynamic extension of the dominant label propagation algorithm [21]. It assigns a belonging factor to each label, where one of the node labels is considered as the main community, and other labels are regarded as the part of the secondary communities [7]. This method can detect both overlapping and non-overlapping communities, but the running time of this is high.

2.2. Information diffusion

There are several models for information diffusion in social networks such as cascade [9], probabilistic diffusion [22] and game theory [23–26]. In cascade model [9], each node can be informed or uninformed. Wang proposed a model where each node informs their neighbours based on the value strength between them [9]. In probabilistic diffusion models [22], each node can be informed based on some probability. Lu et al. proposed a probabilistic diffusion model for weighted network based on community detection [22]. This method first detects community and then initializes source nodes in each community. Each source node informs their neighbours based on two parameters. The first parameter is the weight between the source node and their neighbours and the second is the node degree. In game theory based models [23–26], each player is a selfish agent trying to maximize its utility. Jiang et al. proposed a game theory based model to spread information in dynamic networks [23]. In this method, each node (player) has two strategies. One strategy is forwarding the information and another strategy is not forwarding the information. In this model, each node can update its strategy based on its neighbours strategy. Jiang et al. extended the method to analysing dynamic evolutions of information diffusion [24]. Luo et al. proposed another game theory based model, where it has two players [25]. One player is the infection source and the other is the network administrator. This model defines a safety margin for information diffusion based on the minimum largest distance between the source and distance of infection. The infection source tries to infect as nodes as possible and network administrator attempts to estimate the best source node that probes any nodes in a safety margin. Ok et al. proposed a model based on game theory, where information can be new or old [26]. The coordination gain of new information is much more than old information. Therefore, this method tries to maximize spreading new information.

3. Preliminaries

In this section, we provided notations used throughout the article. Next, the information diffusion mechanism is discussed.

3.1. Notation

Generally, a network can be represented by a graph $G(V,E)$, where V is the set of vertices, and E is the set of edges. Overlapping community detection is the task of finding densely connected overlapping sub graphs in G. Edges can be weighted or unweighted. In this paper, edges are considered unweighted. We define a dynamic network G as a set of sequential graphs $G = \{G_1, G_2, \ldots, G_t\}$, where each graph corresponds to one snapshot. In the dynamic network, snapshots are taken from systems in real networks. In synthetic

networks, snapshots are generated. Symbol T represents the snapshot number starting with 1.

3.2. Information diffusion

The process of effectively spreading information such as new ideas, advertisements, and breaking news is called information diffusion [27]. One of the main parts of information diffusion is social influence. In the real world, individuals are influenced by the actions taken by others [28]. In social networks, nodes also are able to influence other nodes. Finding source nodes, namely the nodes with more influence than other nodes, is important. Another related part is the activation sequence, which is an ordered set of nodes capturing the order in which the nodes receive a label from the seed node. These sequences can affect the performance during the information diffusion process. This process considers a directed graph as G = (N, A) where N is the number of nodes and A is a set of lines where the link (s,d) represents that d get information from c.

Definition 1. (Belonging factor (c,x)). The belonging factor represents the strength of x's membership to the community computed by the following equation [6]:

$$b_t(c,x) = \frac{\sum_{y \in N(x)} b_{t-1}(c,y)}{|N(x)|} \tag{1}$$

Where $b_t(c,x)$ is a function indicating the belonging factor node x to community c in iteration t, and $N(x)$ demonstrates the set of x's neighbours.

4. The proposed method

To detect overlapping communities in dynamic social networks, we propose a novel method. The proposed method consists of three parts. The first part is an initialization part, where each node is given a unique label corresponding to the node number, and the belonging factor of each label is considered as 1. In the second part, the CID model is used. This part takes two states for each node, where the first state is S_1-state representing the amount of affectedness of the node. The second state is S_0-state representing the amount of unaffectedness of the node. Each node can belong to both states, while how much each node belongs of the node to these states is varied. We compute the amount of belonging of the node to S_1 and S_0 states based on value strength measure. The measure is used to compute by following expressions [9]:

$$V_{(i\to j)} = \frac{|C_j \backslash C_i|}{|C_j|} \tag{2}$$

$$V_{(j\to i)} = \frac{|C_i \backslash C_j|}{|C_i|} \tag{3}$$

where C_i denotes the set of i's neighbours and C_j donates the set of j's neighbours. Symbol $(i->j)$ represents the strength value of the node i to j and symbol $(j->i)$ represents the strength value of the node j to i. After computing the value strength measure between each node pair on the graph, we set the amount of belonging of each node to S_1 and S_0 states. For each node, the amount of belonging of the node to S_1-state is computed by:

$$S_{1i} = \frac{\sum_{j \in neighbours(i)} V_{(j\to i)}}{n} \tag{4}$$

where $V_{(j\to i)}$ is the strength value of the node j to i and n is the number of the node i neighbours. The amount of belonging of the node to S_0 state is computed by:

$$S_{0i} = 1 - S_{1i} \tag{5}$$

After specifying the amount of belonging each node to the S_1 and S_0 states, the third part is initiated. This part adds a new property to each node, where it is related to S_1 and S_0 states. For each node, the amount of belonging to these states is computed. The more node belongs to S_1 state, the more affected of the node.

In the third part, one of the nodes is randomly chosen as the node tries to update its label set. Next, the neighbours of the node choose one of the labels whose belonging factor is maximized. Then, each neighbour node specifies its vote based on the belonging factor of the selected label and amount of belonging of the node to S_1 and S_0 States. The vote is computed by:

$$vote_j = S_{0j} * belongingfactor(sl) + S_{1j} * \left(\frac{1 - belongingfactor(sl)}{3}\right) j \in neighbour(i) \tag{6}$$

where *belonging factor(sl)* represents the amount of belonging of the selected label to node j. Symbols S_{1j} and S_{0j} represents the amount of belonging of the node j to S_1 and S_0 states.

Then, the node updates its labels to the label that has the highest vote. After specifying the label that node accepts, it updates the belonging factor of its labels. The third part repeats for each node separately. Finally, the node label that belonging factor is below than the threshold is removed.

4.1. Algorithm description

Algorithm 1 presents the pseudo-code of the proposed method. The algorithm first assigns one label to each node, where the belonging factor of the label is maximized. Next, for each node v, first, the *ObtainNbs()* function extracts neighbours of the node and second, the strength value of each neighbour to node v is calculated. Third, the amount of belonging the node to S_1 and S_0 states is computed based on the node strength to the neighbours.

After computing the amount of belonging the node to S_1 and S_0 states, *Changed Nodes* are placed in random order by the *ShuffleOrder()* function. Next, one of the nodes is chosen from this order, and subsequently the *ObtainNbs()* function extracts neighbours of the node. Then, the *GetLabels()* function extracts candidate labels of each neighbour node. In the following, it creates the candidate label set of the neighbour's labels and subsequently the vote of candidate labels is computed by the *ComputeVote()* function. After that, the label whose vote is maximum is sent to the selected node. The chosen node normalizes its label set by the *Normalize(labelset(i))* function. This function gets the belonging factor of the selected node's labels and set that of the node such that the sum of the belonging factors be 1. Finally, labels with a low belonging factor are removed, and communities are extracted based on similarity in the belonging factor of the labels.

4.2. Time complexity

The proposed method is divided into three parts. In part 1, it requires $O(n)$ for initializing labels, where n is the number of nodes. In part 2, the present writer uses the CID model. In this part, first, for each node, the strength value of the node to the neighbours is computed. This computation requires $O(m)$, where m is the number of edges. Second, the amount of belonging each node to S_1 and S_0 states is calculated. This calculation requires $O(n)$, where n is the number of nodes. Based on the above data, part 2 requires $O(n+m)$. Part 3 is the label propagation part, which requires $O(m)$, where m is the number of edges. From the above fact, it can be concluded that the proposed method in each snapshot requires O(n + m). Since the number of snapshots is a constant T, the overall complexity of community detection for all snapshots is O(n+m). The time complexity of the proposed method and other available methods such as DLPAE, SLPAD and ILCD is in Table 1. Based on

Table 1
The time complexity of various algorithms; CIDLPA; SLPAD; DLPAE; ILCD.

Algorithm	Time Complexity
CIDLPA	O(m+n)
DLPAE	O(m)
SLPAD	O(m)
ILCD	$O(nk^2)$

Table 1, the time complexity of DLAPE and SLPAD is *O(m)*, where m is the number edges. Moreover, the time complexity of ILCD is $O(nk^2)$, where k is the number of communities. Based on the above facts, the time complexity of the proposed method is approximately same as DLPAE and SLPAD, where m is much greater than n.

Algorithm 1: CIDLPA

Input: snapshot G={G_1=<V_1, E_1>,G_2=<V_2, E_2>,…,G_n=<V_n, E_n>}, T
Output: set of communities of G_n
Method:
ΔV={v|v∈G_1}
for ts: =1 T **do** // ts stands for timestamp
  for $v \in \Delta V$
    Node(v).Mem=v;
    Node(v).label.belongingfactor=1
  end for
  ΔV={v | v∈ $G_{(ts+1)}$ ∩ v ∉ G_{ts} } ts≠T
end for
ΔV={v|v∈G_1}
for ts: =1 T do // ts stands for timestamp
  for $v \in \Delta V$ **do**
    neighb(v)=v. ObtainNbs();
    calculate the strength value of the neighb(v) to node v
    calculate the amount of belonging node v to S_1 state
    calculate the amount of belonging node v to S_0 state
  end for
  ΔV={v | v∈ $G_{(ts+1)}$ ∩ v ∉ G_{ts} } ts≠T
end for
ΔE={e|e∈G_1}
for ts: =1 T **do**
  ChangedNodes ={ u,v |(u,v) ∈ΔE}
  V_{old}={v | v∈ G_{ts}∩ v ∉ $G_{(ts+1)}$ } }
  ChangedNodes= ChangedNodes- V_{old}
  for it = 1 : T do // it means iteration
    ChangedNodes.ShuffleOrder();
    for $v \in$ ChangedNodes do
      neighb(v)= v.ObtainNbs();
      candidatelabels=neighb(v).GetLabels();
      ComputeVote(candidatelabels)
      labelset(v).update(candiadatelabels.The maximum vote);
      Nodes(v).Normalize(labelset(v));
  end for
  end for
  remove Nodes(i) labels seen with belonging factor $< r$
  ΔE={e | e ∈ $G_{(ts+1)}$∩ e ∉ G_{ts} }∪{e | e ∈ G_{ts}∩ e ∉ $G_{(ts+1)}$ } ts≠T
end for

5. Experiments

The performance of variations of the newly proposed method and other earlier available methods is evaluated using four real networks and four synthetic networks.

5.1. Real networks

Real networks consist of four networks. The first is a citation network. The second is an Autonomous system-Oregon-1 network. The third is an Amazon Co-Purchasing network (Amazon) and the fourth is the Enron Email network. In the citation network, nodes can only be added. Therefore, the number of nodes in comparison with the immediately preceding snapshot is increased. In other available networks, nodes and edges can be both added and deleted [29]. All real datasets are taken from the SNAP[1] graph library. In this paper, directed edges are considered as undirected edges. We categorized real networks into small and large-scale networks.

5.1.1. Real small networks

5.1.1.1. Arxiv HEP-pH [29]. The Arxiv HEP-pH is a citation network covering articles published between January 1993 and April 2003. It has 34546 nodes and 421578 edges, where the nodes are papers and edges are the citation between papers. Therefore, an edge from node *i* to node *j* can be regarded as citation between papers *i* and *j*. This dataset considered papers and communication of a one month period as one snapshot.

5.1.1.2. Autonomous system-Oregon-1 [30]. The Autonomous System-Oregon-1 is an autonomous system network covering established routers between March 31, 2001 and May 26, 2001. It has 11174 nodes and 23409 edges, where the nodes are routers and edges are traffic flow between them. It has 9 snapshots, where each week is regarded as one snapshot. The summary of two earlier datasets is in Table 2.

5.1.1.3. Enron email [31]. The email network used throughout this paper is the Enron Email dataset. This dataset incorporates email

[1] http://snap.stanford.edu/.

Table 2
Arxiv HEP-pH, Autonomous System and Enron Email datasets Summary.

Data	Arxiv HEP-H	Autonomous Systems – Oregon-1	Enron Email
Number of nodes	34546	11174	36692
Number of edges	421578	23409	367662
Number of snapshots	124	9	90
Type	Directed, Temporal, Unweighted	Undirected, Temporal, Unweighted	Directed, Temporal, Unweighted

Table 3
Amazon Co-Purchasing Network (Amazon) and Enron Email datasets Summary.

Data	Amazon Co-Purchasing Network (Amazon)	DBLP
Number of nodes	403,394	317080
Number of edges	3,387,388	1049866
Number of snapshots	12	365
Type	Directed, Temporal, Unweighted	Undirected, Temporal, Unweighted

Table 4
The core parameters used for the generation of dynamic synthetic networks.

S	N	k	maxd	minc	maxc	O_n	O_{nc}	μ
10	20000	70	180	60	150	400	3	0.3

Table 5
Custom parameters used for the generation of dynamic synthetic networks.

Birth/Death		Expand/Contract		Merge/Split		Switch
Birth	Death	Expand	Contract	Merge	Split	p
15	15	15	15	15	15	0.3

exchanged between the Enron Email Corporation over 15-years, with two months being one snapshot. In this network, nodes are Enron employees and edges are the emails exchanged between them. The summary of two earlier datasets is in Table 3.

5.1.2. Real large networks

5.1.2.1. Amazon co-purchasing network (Amazon) [32]. The Amazon network is a large-scale network representing the co-purchasing of commodities sold by Amazon.com. We take snapshots of the year 2003 on the Amazon network, where each month is considered as one snapshot. It contains 403,394 nodes and 3,387,388 edges. The average degree of nodes is 7.8. The nodes on the network represent the commodities such as books, CD, DVD and other commodities sold by Amazon. If a product i is frequently co-purchased with a product j, the graph contains a directed edge from i to j.

5.1.2.2. DBLP [32]. DBLP is a computer science bibliography providing a comprehensive list of research papers in computer science. In this dataset, a co-authorship network is constructed, where two authors are connected if they publish at least one paper together. Unlike the citation network, the DBLP network includes ground-truth communities. It considers papers published between 2000 and 2009. It contains 317,080 nodes and 1,049,866 edges, where every ten days are considered as one snapshot. The summary of the DBLP and Amazon Co-Purchasing Network (Amazon) datasets is presented in Table 3.

5.2. Synthetic networks

The performance of the proposed method and earlier available methods is also evaluated in four synthetic networks. These networks are generated through methods introduced by [33]. The core parameters for generating these networks is described by [34]. These parameters are tabulated in Table 4. In Table 4, symbol s is set to 10, which means, the dataset consists of 15 networks snapshots. Symbol n donates the number of nodes. Symbols k and $maxd$ represent the average and max degree of nodes respectively. The average degree of a graph G is a measure of how many edges are in set E compared with the number of vertices in set V.

Symbols $minc$ and $maxc$ represent the maximum and minimum size of the community. Symbols O_n and O_{nc} specify the number of overlapping nodes and the number of each overlapping node community. Symbols O_n and O_{nc} are set to 400 and 3 respectively. This means 400 nodes are overlapping and the nodes belong to three communities. Symbol μ is the mixing parameter, which has a remarkable effect on the performance of the dynamic synthetic network generation [33]. We set this symbol to 0.3, which is the moderate value.

Custom parameters for generating four dynamic networks are in Table 4. According to the method described in [33], the communities in the dynamic networks mainly include four evolution events: i) expansion and contraction, ii) merging and splitting, iii) birth and death and iv) switching nodes. Each evolution event corresponds to one of these dynamic synthetic networks. In Table 5, parameters except p specify the number of corresponding events per timestamp and p is the probability that a node switches community membership between timestamps.

5.3. Evaluation

This paper applies the modularity measure for accuracy evaluation as a result of the lack of ground-truth communities in real networks [35]. The original modularity handles disjointed communities [35], while in this paper, communities can share members. We adopt the extended version of modularity [36] that can deal with the overlapping community structures as well:

$$Q^c_{Ov} = \frac{1}{2m}\sum_{c\in C}\sum_{i,j\in V}[A_{i,j} - \frac{k_i k_j}{2m}]\beta_{ic}\beta_{jc} \tag{7}$$

$$\beta_{ic} = \frac{k_{ic}}{\sum_{j\in c} k_{jc}} \tag{8}$$

where symbols k_i and k_j are the degree of vertexes i and j respectively. Symbol m is the number of edges, and A_{ij} is the adjacency matrix. Symbols β_{ic} and β_{jc} express the strength of belonging nodes i and j to community c respectively. Moreover,k_{ic} is the total weight of links from the node i to community c and k_{jc} is the total weight of links from the node j to community c. To evaluate synthetic networks and DBLP network accuracy; the Normalized Mutual Information (NMI) measure is employed. The NMI measure compares detected communities with ground-truth communities and computes their similarity. The higher the similarity between these communities, the higher the NMI value will be. When ground-truth communities of the network exist, the NMI becomes the most robust and precise parameter to measure accuracy. In this paper, we

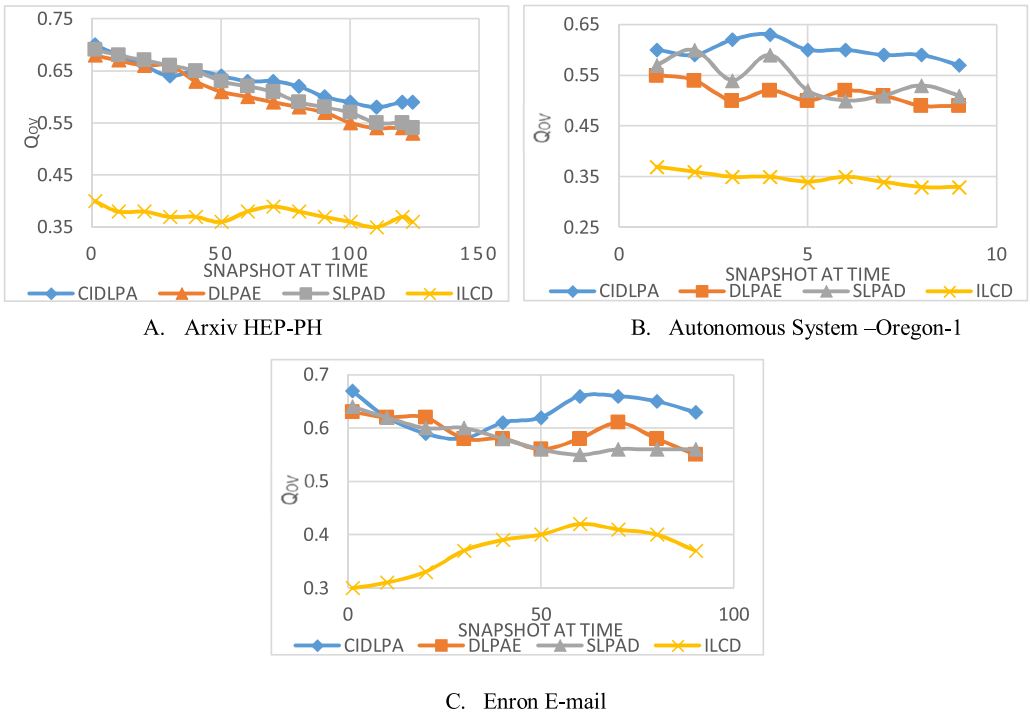


Fig. 1. The modularity values achieved by various algorithms on Arxiv HEP-pH, Autonomous System–Oregon-1 and Enron-Email networks; CIDLPA; SLPAD; DLPAE; ILCD.

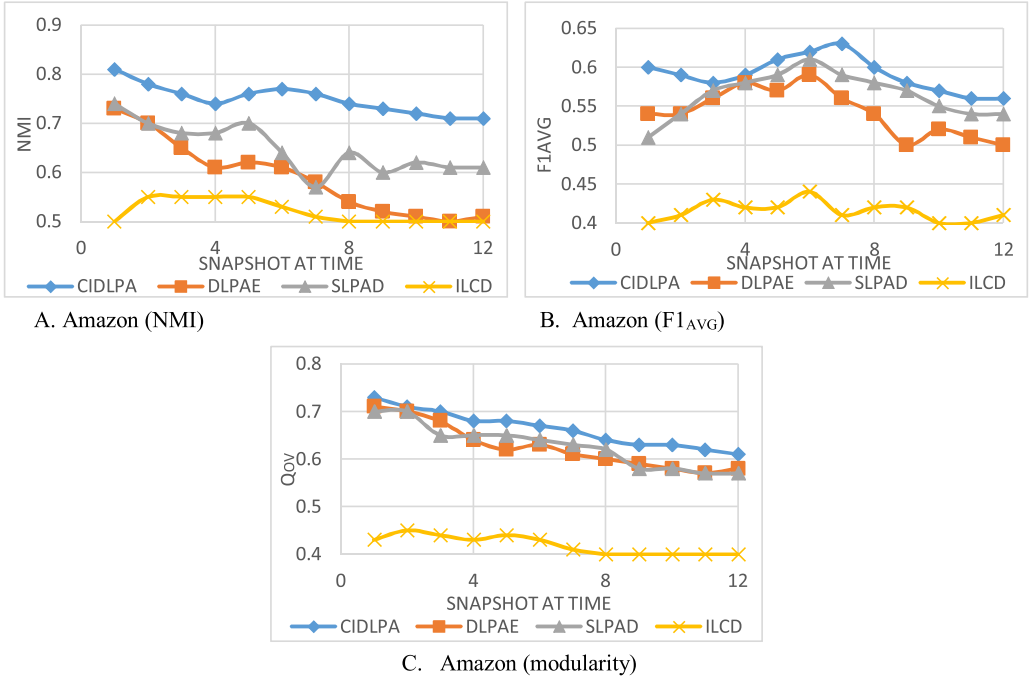


Fig. 2. The NMI, $F1_{AVG}$ and modularity values achieved by various algorithms on Amazon Co-Purchasing Network (Amazon); CIDLPA; SLPAD; DLPAE; ILCD.

aim to discover overlapping communities. The overlapping extension of this measure is defined as [37]:

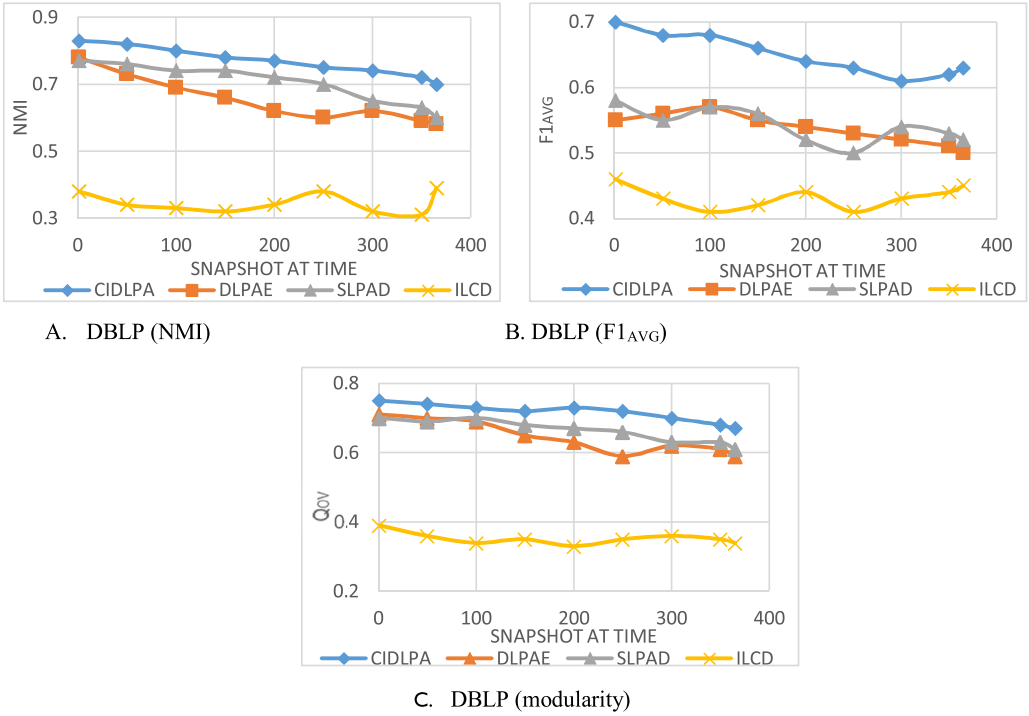
$$NMI(X, Y) = 1 - \frac{H(X|Y) + H(Y|X)}{2} \tag{9}$$

where X and Y are random variables. H(X|Y) is the normalized conditional entropy of a cover X with respect to Y, and H(Y|X) is the normalized conditional entropy of a cover Y with respect to X. H(X|Y) is defined as [37]:

$$H(X|Y) = \frac{1}{|C'|}\sum_{K}\frac{H(X_k|Y)}{H(X_k)} \tag{10}$$

$$H(X_k|Y) = min_{l \in \{1,2,\ldots,|C''|}H(X_k|Y_l) \tag{11}$$

Symbol $H(X_k|Y_l)$ is the conditional entropy of a cluster X_k given Y_l and symbol $H(X_k|Y)$ is the conditional entropy of a cluster X_k given Y. C' and C'' are clusters. $H(Y|X)$ is defined in the same way. This



A. DBLP (NMI) B. DBLP ($F1_{AVG}$)

C. DBLP (modularity)

Fig. 3. The NMI, $F1_{AVG}$ and modularity values achieved by various algorithms on DBLP Network (Amazon); CIDLPA; SLPAD; DLPAE; ILCD.

measure is applied on the DBLP and synthetic networks. The F1-score is another measure used in order to examine with increased precision.

The NMI focus on the overall measure, while the F1-score pays attention to the node level. With the F1-score, we consider the overlapping community detection as binary detection [38]. The range of F1-score values is between 0 and 1. $F1(C_1, C_2)$ is the harmonic mean of precision and recall between

node-sets C_1, C_2. Node-set C_1 is related to grand-truth cover and C_2 is related to the evaluated cover. We need to determine $C_i \in C_1$ corresponds to $C_j \in C_2$. We define $F1_{AVG}(C_1, C_2)$ to be the average of $F1(C_1, C_2)$ of the best-matching ground-truth community to each detected community, and the F1-score of the best-matching detected community to each ground-truth community [38]:

$$precision(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1|} \tag{12}$$

$$recall(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_2|} \tag{13}$$

$$F1\left(C_i, C_j\right) = \frac{2.precision\left(C_i, C_j\right).recall\left(C_i, C_j\right)}{recall\left(C_i, C_j\right) + precision\left(C_i, C_j\right)} C_i \in C_1, C_j \in C_2 \tag{14}$$

$$F1_{AVG}(C_1, C_2) = \frac{1}{2|C_1|} \sum_{C_i \in C_1} maxiF1(C_i, C_2) + \frac{1}{2|C_2|} \sum_{C_j \in C_2} maxjF1(C_j, C_1) \tag{15}$$

where Symbols C_1 and C_2determine evaluated and ground-truth cover node-sets respectively. Measure $precision\,(C_1, C_2)$ is the number of correctly detected overlapping nodes divided by the total number of detected overlapping nodes and$recall\,(C_1, C_2)$ is the number of correctly detected overlapping nodes divided by the true number of detected overlapping nodes.

5.4. Results

The results of implementing CIDLPA, SLPAD [20], DLPAE [7] and ILCD [18] on all available datasets including real and synthetic networks are shown in Figs. 1–7 respectively. We chooses SLPAD and DLPAE because both are able to detect overlapping communities based on the label propagation approach in dynamic social networks. Moreover, it chooses ILCD because it tries to find overlapping communities in dynamic social networks.

5.4.1. Results on real networks

First, we conduct experiments on five real-world networks: Arxiv HEP-pH, Enron Email and Autonomous System–Oregon-1, Amazon and DBLP. NMI and $F1_{AVG}$ can be used when ground truth communities of the datasets is available. In Arxiv HEP-pH, Enron Email and Autonomous System–Oregon-1 datasets, the ground truth of these three datasets are unavailable. Therefore, we were not able to use NMI and $F1_{AVG}$ to evaluate the accuracy of the proposed method and other available algorithms, and we must use the modularity [37] to evaluate method performance.

5.4.1.1. Results on real small networks. Arxiv HEP-pH, Enron Email and Autonomous System–Oregon-1 datasets are small networks. The accuracy of our algorithm is compared with SLPAD, DLPAE, and ILCD using measurement modularity. The results are given in Fig. 1. We can observe that our algorithm performs better than the other three algorithms on the above three real datasets. Obviously, this is the benefit from considering the CID model in the proposed method and using two sets (S_1 and S_0). Moreover, as can be seen from Fig. 1, the weakest method is ILCD. That is why it needs prior knowledge. Based on the above facts, it can be concluded that the proposed method performs better based on the modularity measure compared to other available methods in small networks.

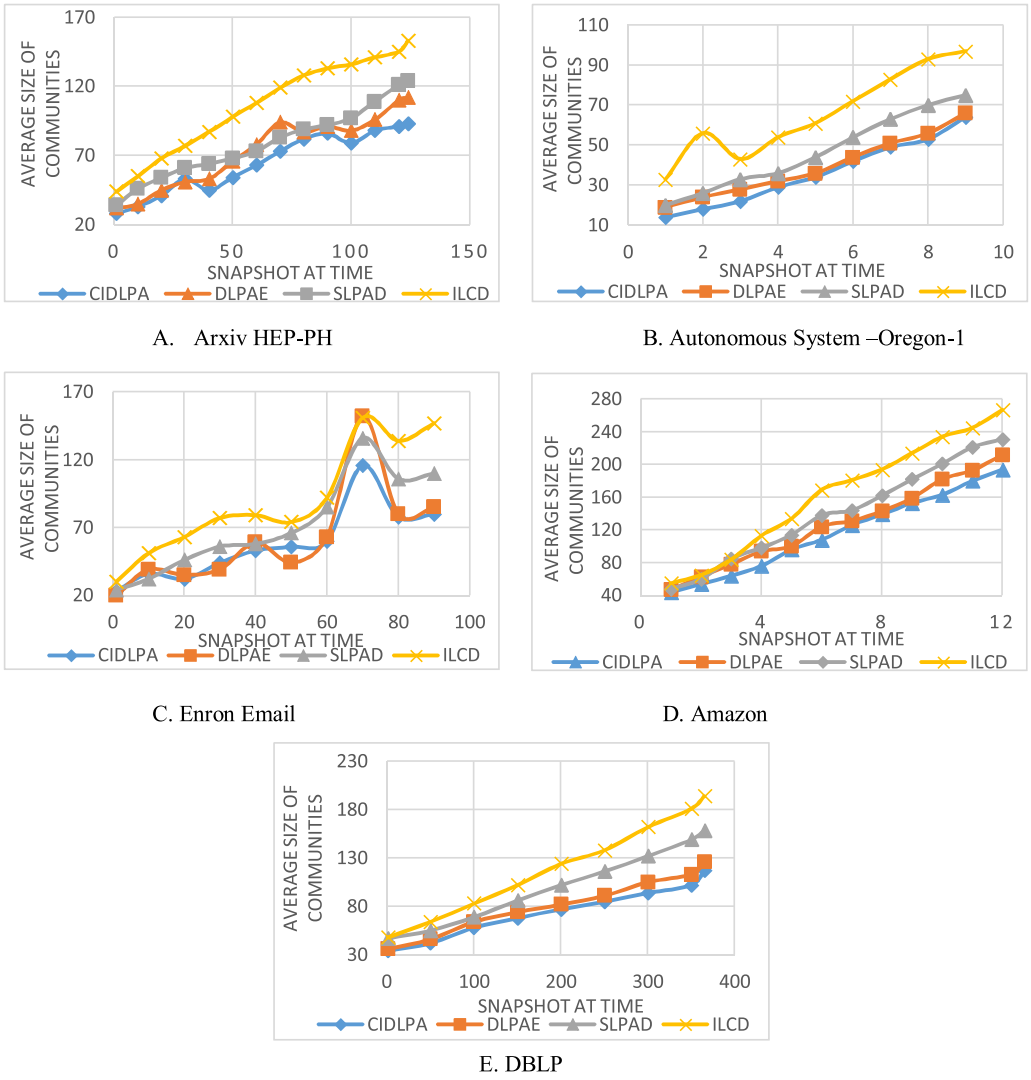


Fig. 4. The average size of communities achieved by various algorithms on real networks; CIDLPA; SLPAD; DLPAE; ILCD.

5.4.1.2. Results on real large-scale networks. The accuracy of our algorithm is compared with SLPAD, DLPAE, and ILCD using NMI, $F1_{AVG}$ and modularity in the Amazon dataset and DBLP. The results are given in Figs. 2 and 3. It can be observed that CIDLPA gains the obvious advantage over the other three algorithms based on the NMI and modularity measure, which indicates that CIDLPA is appropriate to the discovery communities in dynamic networks. Since the proposed method takes node roles more than other available methods, the method performs better than the methods by considering $F1_{AVG}$. Moreover, as the Amazon network and DBLP are large-scale networks, it also represents the scalability of the proposed algorithm. Based on the above facts, it can be concluded that the proposed method performs better based on NMI, $F1_{AVG}$ and modularity measures than other available methods in large-scale networks.

5.4.1.3. Results on real small and large-scale networks. The average of modularity of the proposed method and other available methods in real networks is in Table 6. The results show that the proposed method gains better modularity in Amazon and DBLP than other datasets. Therefore, it can be concluded that the proposed method achieves higher modularity in the large-scale networks than small networks. That is why that the more node in large-scale networks than small networks can vote for candidate labels.

Table 6
The average modularity achieved by various algorithms on Real networks; CIDLPA; SLPAD; DLPAE; ILCD.

Datasets	Methods			
	CIDLPA	DLPAE	SLPAD	ILCD
Arxiv HEP-pH	0.629	0.601	0.614	0.373
Autonomous System–Oregon-1	0.598	0.513	0.541	0.346
Enron E-mail	0.629	0.591	0.583	0.37
Amazon	0.663	0.625	0.628	0.419
DBLP	0.716	0.663	0.643	0.352

The average size of communities and running time of the proposed method is compared with SLPAD, DLPAE, ILCD in real networks. The results are given in Figs. 4 and 5. It can be observed that the average size of communities on average in CIDLPA is lower than other available methods. Based on the above facts, it can be concluded that the proposed method decreases the chance of creating monster communities. Moreover, the proposed method is slightly slower than other available methods. That is why the method has more computation time than these methods.

5.4.2. Results on synthetic networks

We conducted experiments on four generated dynamic networks. The comparison of NMI in the proposed method with other

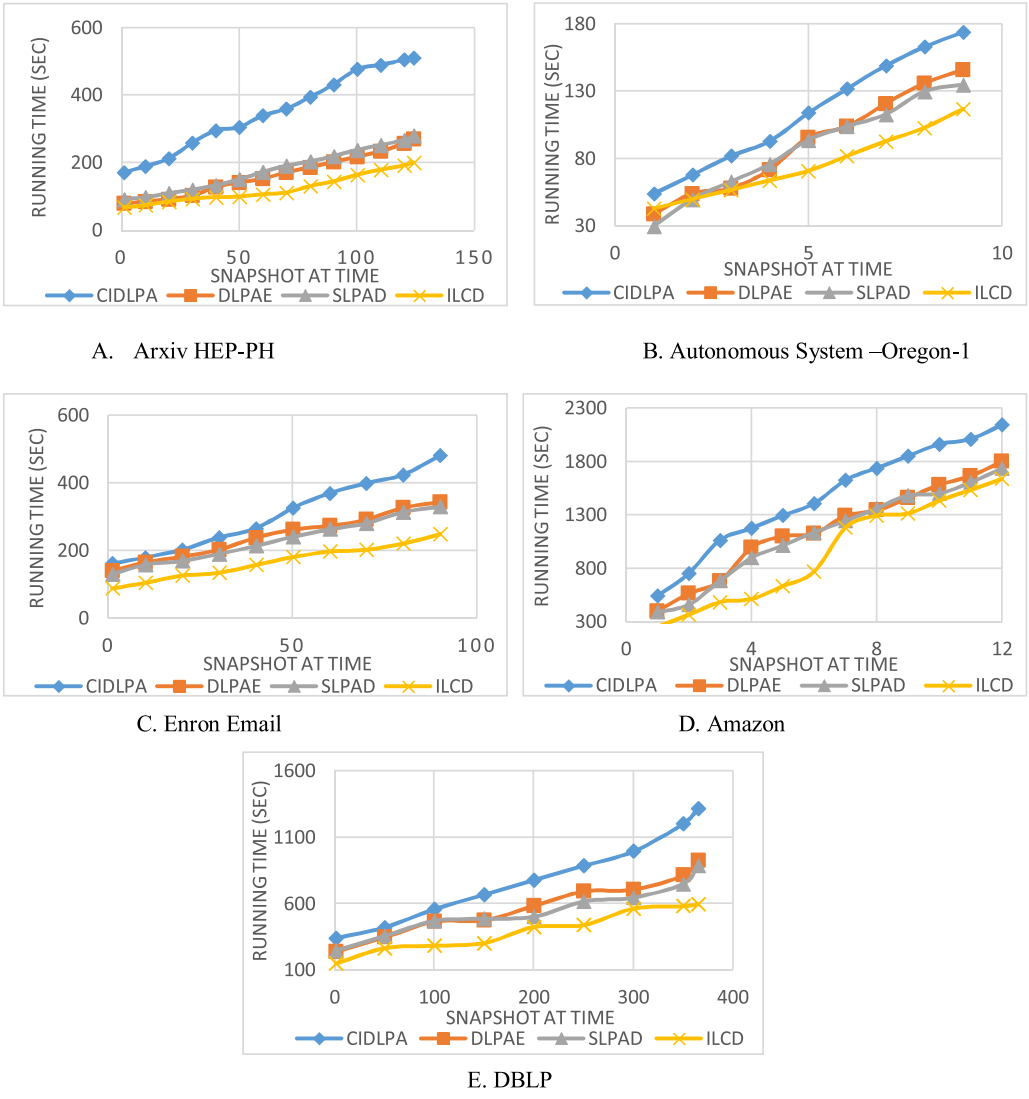


Fig. 5. The running time values achieved by various algorithms on real networks; CIDLPA; SLPAD; DLPAE; ILCD.

Table 7
The average NMI achieved by various algorithms on Synthetic networks; CIDLPA; SLPAD; DLPAE; ILCD.

Datasets	Methods			
	CIDLPA	DLPAE	SLPAD	ILCD
Birth/Death	0.626	0.591	0.583	0.383
Expand/Contract	0.567	0.53	0.533	0.372
Merge/Split	0.521	0.525	0.482	0.395
Switch	0.538	0.518	0.503	0.511

Table 8
The average $F1_{AVG}$ achieved by various algorithms on Synthetic networks; CIDLPA; SLPAD; DLPAE; ILCD.

Datasets	Methods			
	CIDLPA	DLPAE	SLPAD	ILCD
Birth/Death	0.672	0.591	0.6	0.462
Expand/Contract	0.649	0.601	0.58	0.445
Merge/Split	0.604	0.615	0.574	0.436
Switch	0.623	0.589	0.583	0.426

available methods on four synthetic datasets is plotted in Fig. 6. Moreover, the average NMI of the method and other available method is tabulated in Table 7. Fig. 6 shows that in all events except for Merge and Split events, the proposed method achieves higher NMI compared to other available methods. However, based on [39], Merge and Split events compose only a few percent of occurred events. Furthermore, based on data from Table 7, the proposed method gains more NMI in Birth\Death events in comparison to other events. That is why the method concentrates on new nodes more than other available methods.

The comparison of $F1_{AVG}$ of the proposed method with other available methods on four synthetic datasets is plotted in Fig. 7. Moreover, the average of $F1_{AVG}$ of the proposed method and other available method is in Table 8. Fig. 7 compares the $F1_{AVG}$ by different methods. It shows that CIDLPA achieved more $F1_{AVG}$ in three of four datasets, followed by DLAPE and SLPAD. That is because, in our method, we consider node strength values. Furthermore, based on data from Table 8, the proposed method gains more $F1_{AVG}$ in all events except for Merge/Split in comparison with other available methods.

The average size of communities and running time of the proposed method is compared with SLPAD, DLPAE, and ILCD in synthetic networks. The results are given in Tables 9 and 10. It can be observed that like real networks, the proposed method is slightly slower than other available methods. Moreover, the

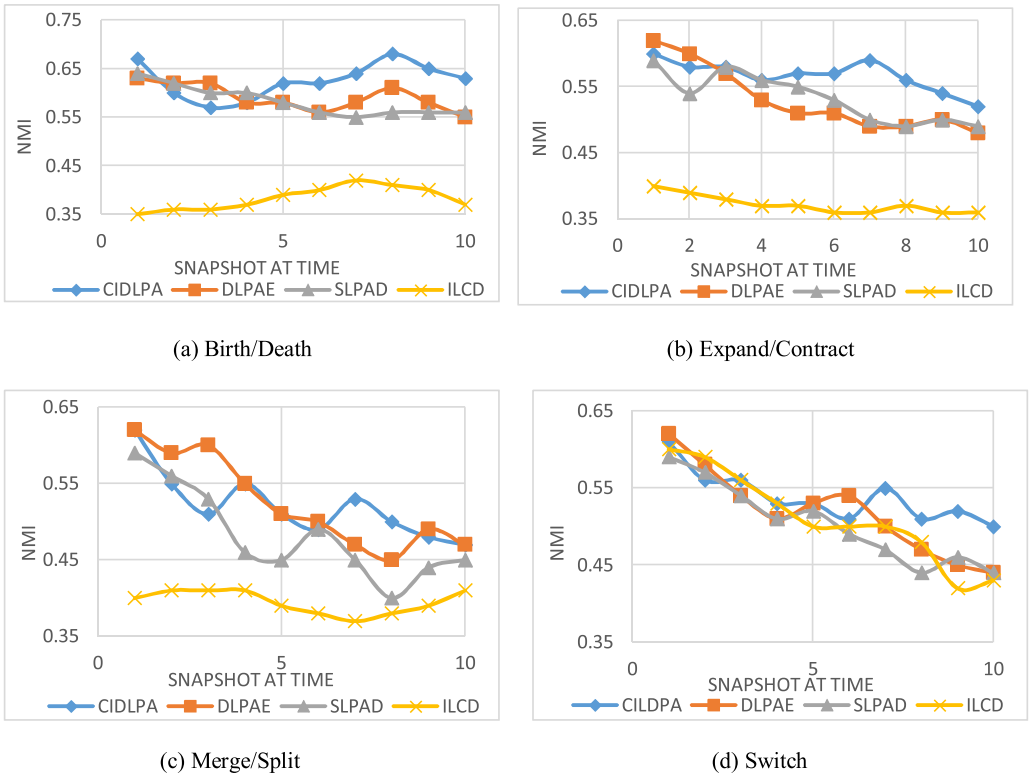


Fig. 6. The NMI values achieved by various algorithms on Synthetic networks; CIDLPA; SLPAD; DLPAE; ILCD.

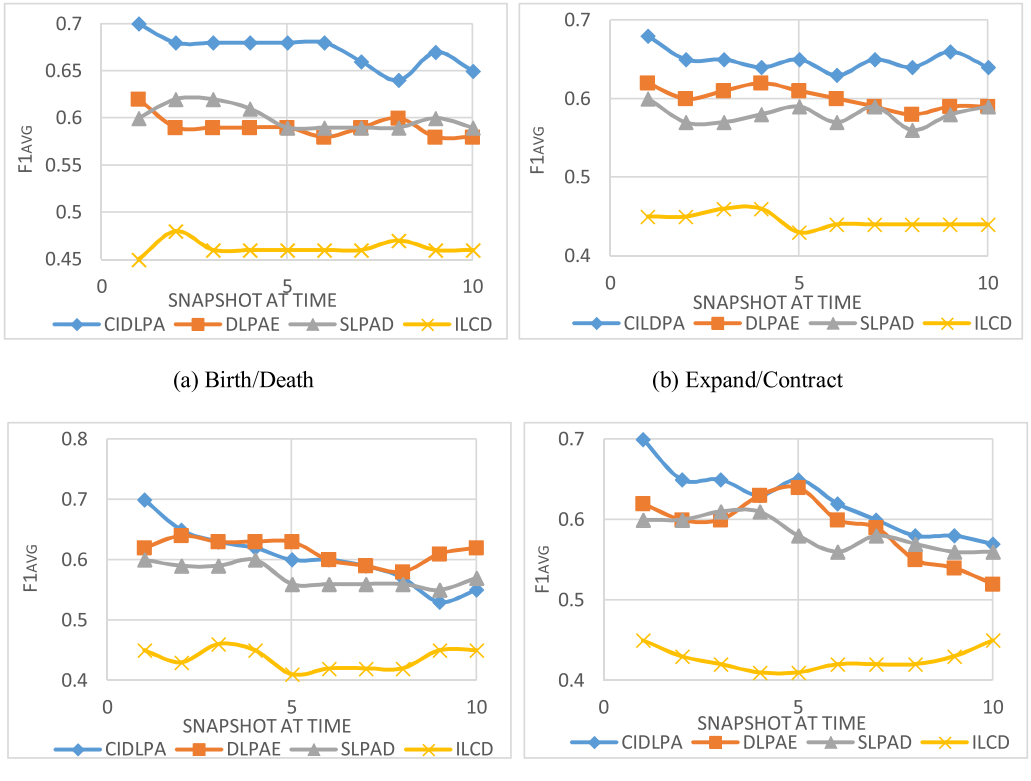


Fig. 7. The values of $F1_{AVG}$ achieved by various algorithms on Synthetic networks; CIDLPA; SLPAD; DLPAE; ILCD.

Table 9
The average size of communities achieved by various algorithms on synthetic networks; CIDLPA; SLPAD; DLPAE; ILCD.

Datasets	Methods			
	CIDLPA	DLPAE	SLPAD	ILCD
Birth/Death	86	81	76	45
Expand/Contract	97	85	75	49
Merge/Split	71	65	62	43
Switch	78	68	63	44

Table 10
The average running time achieved by various algorithms on synthetic networks; CIDLPA; SLPAD; DLPAE; ILCD.

Datasets	Methods			
	CIDLPA	DLPAE	SLPAD	ILCD
Birth/Death	656	581	583	390
Expand/Contract	667	595	585	380
Merge/Split	621	535	523	363
Switch	638	528	513	354

method detects smaller community compared to other available methods.

6. Conclusion

In this paper, we proposed a novel cascade information diffusion based label propagation method in order to detect overlapping communities in dynamic social networks. According to the proposed method, each node can belong to two states, which distinguish nodes of either based on its affectedness or unaffectedness. This difference between nodes reduced the chance of creating monster communities in dynamic networks. We compared the proposed method with three available methods for overlapping community detection in dynamic social networks. This comparison is applied to four real and synthetic networks. We used the synthetic networks due to lack of grand-truth communities and evaluated the effect of various events on the performance of the proposed method. The results showed that the proposed method increased the accuracy of overlapping community detection in dynamic social networks. Therefore, the proposed method produces more meaningful communities than other available methods. Detecting communities in dynamic networks is very challenging and discovering dynamic communities is still in its infancy and can be addressed by using methods such as our proposed method in the future.

References

[1] C.C. Aggarwal, An Introduction to Social Network Data Analytics, Springer, US, 2001, pp. 1–15.
[2] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[3] S. Fortunato, Community detection in graphs, Phys. Rep. (2010) 74–174.
[4] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.
[5] J. Xie, B.K. Szymanski, X. Liu, SLPA. Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, in: 11th International Conference on Data Mining Workshops (ICDMW), IEEE, 2011, pp. 344–349.
[6] S. Gregory, Finding overlapping communities in networks by label propagation, New J. Phys. 12 (10) (2010) 103018.
[7] K. Liu, J. Huang, H. Sun, M. Wan, Y. Qi, H. Li, Label propagation based evolutionary clustering for detecting overlapping and non-overlapping communities in dynamic networks, Knowl.-Based Syst. 89 (2015) 487–496.
[8] X. Deng, Y. Wen, Y. Chen, Highly efficient epidemic spreading model based LPA threshold community detection method, Neurocomputing 210 (2016) 3–12.
[9] J. Wang, C. Jiang, T.Q. Quek, X. Wang, Y. Ren, The value strength aided information diffusion in socially-aware mobile networks, IEEE Access 4 (2016) 3907–3919.
[10] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, Bell Syst. Tech. J. 49 (2) (1970) 291–307.
[11] 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, ACM, 2007, pp. 824–833.
[12] A. Clauset, M.E.J. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 1–6.
[13] K. Kuzmin, S.Y. Shah, B.K. Szymanski, Parallel overlapping community detection with SLPA, in: International Conference on Social Computing (SocialCom), IEEE, 2013, pp. 204–212.
[14] J. H. Sun, J. Liu, G. Wang, Z. Yang, Q. Song, X. Jia, CenLP, a centrality-based label propagation algorithm for community detection in networks, Physica A 436 (2015) 767–780.
[15] N.P. Nguyen, T.N. Dinh, Y. Xuan, M.T. Thai, Adaptive algorithms for detecting community structure in dynamic social networks, in: IEEE Conference on Computer Communications (INFOCOM), IEEE, 2011, pp. 2282–2290.
[16] Y.R. Lin, Y. Chi, S. Zhu, H. Sundaram, B.L. Tseng, Facetnet: a framework for analyzing communities and their evolutions in dynamic networks, in: Proceedings of the 17th International Conference on World Wide Web, ACM, 2008, pp. 685–694.
[17] J. Xie, M. Chen, B.K. Szymanski, T. Labelrank, Incremental community detection in dynamic networks via label propagation, in: Proceedings of the Workshop on Dynamic Networks Management and Mining, ACM, 2013, pp. 25–32.
[18] R. Cazabet, F. Amblard, C. Hanachi, Detection of overlapping communities in dynamical social networks, in: International Conference on Social Computing (SocialCom), IEEE, 2010, pp. 309–314.
[19] A. Lancichinetti, F. Radicchi, J.J. Ramasco, S. Fortunato, Finding statistically significant communities in networks, PLoS One 6 (4) (2011) e18961.
[20] N. Aston, J. Hertzler, W. Hu, Overlapping community detection in dynamic networks, J. Softw. Eng. Appl. 7 (10) (2014) 872.
[21] S. He-Li, H. Jian-Bin, T. Yong-Qiang, S. Qin-Bao, L. Huai-Liang, Detecting overlapping communities in networks via dominant label propagation, Chin. Phys. B 24 (2015) 551–559.
[22] Z. Lu, Y. Wen, W. Zhang, Q. Zheng, G. Cao, Towards information diffusion in mobile social networks, IEEE Trans. Mob. Comput. 15 (5) (2016) 1292–1304.
[23] C. Jiang, Y. Chen, K.R. Liu, Graphical evolutionary game for information diffusion over social networks, IEEE J. Sel. Top. Signal Process. 8 (4) (2014) 524–536.
[24] C. Jiang, Y. Chen, K.R. Liu, Evolutionary dynamics of Information diffusion over social networks, IEEE Trans. Signal Process. 62 (17) (2014) 4573–4586.
[25] W. Luo, W.P. Tay, M. Leng, Infection spreading and source identification: a hide and seek game, IEEE Trans. Signal Process. 64 (16) (2016) 4228–4243.
[26] J. Ok, Y. Jin, J. Shin, Y. Yi, On maximizing diffusion speed over social networks with strategic users, IEEE/ACM Trans. Netw. 24 (6) (2016) 3798–3811.
[27] F. Wang, H. Wang, K. Xu, Diffusive logistic model towards predicting information diffusion in online social networks, in: Distributed Computing Systems Workshops (ICDCSW), IEEE, 2012, pp. 133–139.
[28] A. Anagnostopoulos, R. Kumar, M. Mahdian, Influence and correlation in social networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 7–15.
[29] J. Gehrke, P. Ginsparg, J. Kleinberg, Overview of the 2003 KDD cup, ACM SIGKDD Explor. Newslett. 5 (2) (2003) 149–151.
[30] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2005, pp. 177–187.
[31] B. Klimmt, Y. Yang, The enron corpus: a new dataset for email classification research, in: Proceedings of the European Conference on Machine Learning (ECML), Springer Berlin Heidelberg, 2004, pp. 217–226.
[32] J. Yang, J. Leskovec, Defining and evaluating network communities based on ground-truth, Knowl. Inf. Syst. 42 (1) (2015) 181–213.
[33] D. Greene, D. Doyle, P. Cunningham, Tracking the evolution of communities in dynamic social networks, in: International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE, 2010, pp. 176–183.
[34] A. Lancichinetti, S. Fortunato, Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities, Phys. Rev. E 80 (1) (2009) 016118.
[35] M. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. U. S. A. 103 (23) (2006) 8577–8582.
[36] D. Chen, M. Shang, Z. Lv, Y. Fu, Detecting overlapping communities of weighted networks via a local algorithm, Physica A 389 (19) (2010) 4177–4187.
[37] A. Lancichinetti, S. Fortunato, J. Kertesz, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. 11 (3) (2009).
[38] J. Yang, J. Leskovec, Community-affiliation graph model for overlapping network community detection, in: International Conference on Data Mining (ICDM), IEEE, 2012, pp. 1170–1175.
[39] M. Takaffoli, J. Fagnan, F. Sangi, O.R. Zaïane, Tracking Changes in Dynamic Information Networks, Computational Aspects of Social Networks (CASoN), IEEE, 2011, pp. 94–101.

Mohammad Sattari is a Ph. D. Student at the Faculty of Computer Engineering, University of Isfahan since 2013. He received her M. Sc. degree in computer engineering with honors. His Ph. D. thesis is focused on community detection in dynamic social networks. His main interests include social network analysis and data mining.

Kamran Zamanifar is Associate Professor at the University of Isfahan since 1996. He received his M. Sc. in electrical and electronic engineering from the Faculty of Engineering, University of Tehran, Iran. He also received his Ph. D. in computer science (parallel and distributed systems) from the School of Computer Studies, University of Leeds, England (1992–1996). He is a member of various scientific committees such as Management Board of Computer Society of Iran, Iranian Association of Electrical and Electronic Engineers and Annual International CS Computer Conferences. He is also the reviewer of many scientific journals. His main interests include parallel and distributed systems, concurrent systems and high performance computing. He has published several books and some journal and conferences papers.