A New Framework based on Learning Automata for User Community Detection in Social Networks

Rahebeh Mojtahedi Saffari[1,*] , Hassan Rashidi[2]

[1]Department of Computer Engineering, Islamic Azad University, Lahijan Branch, Lahijan, Iran
[2]Department of Mathematics and Computer Science, Allameh Tabataba'i University, Tehran, Iran

Abstract

Recently, social networks provide some rich resources of heterogeneous data which their analysis can lead to discover unknown information and relations within such networks. Users in online social networks tend to form community groups based on common location, interests, occupation, etc. Hence, communities play special roles in the structure–function relationship. Therefore, detecting significant and densely connected user communities from social networks has become one of the major challenges that help to understand some behavioral characteristics of users in social networks. Moreover, discovered communities can be a way to describe and analyze such networks. Most recent works on user community detection has focused on analyzing either user-friendship structure or user-generated contents but not both at the same time. In this paper, we propose a new framework based on distributed learning automata for detecting user community that considers user-friendship structure and user content information simultaneously. Finally we have evaluated our framework on the Twitter dataset. The evaluation results indicate that this framework is able to discover substantial user communities in which there are dense relationships among members.

Keywords: *Social network, user Community detection, Distributed learning automata, User-topic modeling.*

1. Introduction

Community structure is an important feature in complex networks. In recent years, the researches of looking for community structures in the network attracted more and more subject areas. Community discovery technology is widely applied in some specific areas such as biology, physics, computer science, business activity and sociological fields. For example, it can be used to detect criminal gangs, find potential customers and personalized services and so on. Social network is another example of complex networks. Web 2.0 technology has enabled massive online social networks and made sharing of user-generated contents easy and almost costless. Two-thirds of Americans now use Facebook, Twitter, Myspace, andother social media[1] sites; and 43% are visiting these sites more than once a day. By May 2010, social networks have become more popular than search engines in U.K., accounting for 11.88% of all U.K. Internet visits. Usually, a social network involves multiple types of relations among different social actors. For instance, on Twitter, a user can specify whom to follow to construct an explicit friendship network. At the same time, this user's posted tweets provide important clues about her interests and such interests across the user community can be used to derive implicit "similarity" relationships among these users. Community discovery not only helps to understand the structure and function of the network structure, but also provides important technical means for the transformation of the network and analysis of network characteristics. The network embedding multiple types of relations, either explicit or implicit, is called a multi-relational network. Studying multiple relationships is gaining momentum in the literature recently. In social network, not every node belongs to only one community. On the contrary, there are "overlapping nodes" which belong to multiple communities, and the communities which have overlapping nodes with other communities are called "overlapping communities." Overlap is very common. For example, in a social network, a person can participate in basketball club and football club at the same time, which means that a person belongs to multiple user communities. Because of overlap nodes, Overlapping user communities probably contact with each other, so overlapping nodes are important "bridge" for connecting different communities. Overlapping community detection is of important guiding significance for research of network topology. Discovering overlapping user communities may assist the setup of efficient recommender systems for targeted marketing, improving the quality of social information retrieval, among others [1,2,3,4,5]. For instance, Nie et al. [4] utilized the relevance of communities to improve web page ranking. In [1], the authors investigated how consumers take advantage of virtual communities as social and information networks, and how this influences their decision making. The identified user communities can also help understand the structural properties of the social network and find the influential users about certain topics, which in turn will help users locate the latent friends they may be interested in.

Most recent works on user community detection has focused on analyzing either user friendship networks[6,7,8]or user-generated contents[9,10]but not both at the same time. The former techniques usually ignore the content generated by users. However, intuitively, two users who have posted similar contents might share common interest and join the same communities, even if no explicit friendship connection exists between them. On the other hand, the latter strategies do not take the friendship connections among users into consideration. Such explicit friendship networks can provide important clues to community discovery.

In this paper, we focus on the problem of discovering user communities from multi-relational networks. We present a unified framework based on distributed learning automata, which combines the Latent Dirichlet Allocation (LDA; Blei et al., 2003) topic model with social network analysis (SNA). LDA model, which deals with document content modeling, while the SNA methods focus on user friendship networks. This

*Corresponding Author: Rahebeh Mojtahedi saffari

framework considers structural and contextual information of users simultaneously.

The rest of the paper is organized as follows. The literature review is presented in Section 2. Section 3 presents in detail the problem definition and our proposed framework. The empirical analysis is conducted in Section 4. Finally, Section 5 concludes the paper with a summary and discussion of the future work.

2.Literature Review

In this paper, with reviewing many other methods about user community detection, we have classified them in following categorizations:

A. Non-overlapping and structure-based methods:

This methods construct network structures among users and then split the network into different sub-networks.

There is not any node which belongs to multiple sub-networks.

1) SCD:stability community detection

This method works by first enriching the input network with the mutual relationship estimation of all links and then discovering stable communities using a lumped Markov chain model[41]. SCD has the advantage of handling the real model of OSNs with weighted reciprocity relationships. Procedure of community detection is as follows:

1. an estimation which provides helpful insights into the stability of links in the input network.
2. exploring an vital connection between the persistence probability of a community at the stationary distribution and its local topology, which is the fundamental mathematical theory to develop the SCD method.

2) Hierarchical clustering

This technique[11] falls into two principal methods: divisive and agglomerative. Indeed, the divisive clustering is a top-down approach which starts by only one cluster which group all nodes and then separates the most dissimilar clusters until all objects are concepts leaves. The agglomerative clustering is a bottom up approach where each object presents a cluster; iteratively combine the nearest objects in clusters until reaching the desired number of clusters.

3) Newman and Girvan's method (GN)

The GN technique [12] is a top-down hierarchical clustering, and based on the shortest path betweenness measure. Actually, the betweenness of an edge in an undirected graph designates the number of shortest paths between pairs of nodes that run through this edge. The principle of this method is to remove edge with the highest betweenness centrality and to recalculate for each iteration this value. This method also propose a measure for the strength of the community structure, which gives us an objective metric for choosing the number of communities into which a network should be divided. This measure is called modularity. The above method is based on the optimization of the modularity function, which is written as:

$$a_i = \sum p_j \, e_{ij} \qquad (1)$$

$$Q = \sum_i e_{ii} - a_i^2 \qquad (2)$$

Q is a measure that presents the difference between the fraction of edges intra-community and the same expected value for a random fraction of edges.

B. Overlapping and structure based methods

1) NLEM method

Newman and Leicht (2007) have proposed a method (NLEM) based on a mixture model and the expectation maximization (EM) technique (Dempster et al. 1977). Their model assumes that n nodes of the network fall into k communities. The nodes are encoded as an adjacency matrix A where Aij = 1 if there is a directed relationship from node i to node j.

The group of node i is indicated by gi, pr is the fraction of nodes belonging to group r; and hri is the probability that there is a directed relation from group r to node i. From Newman and Leicht's model and through EM technique, they derive three equations that maximize the log-likelihood that the model fits the adjacency matrix A of the network which are the following:

$$q_{ir} = p(g_i = r|A, \pi, \theta) = \frac{\pi_r \prod_j \theta_{rj}^{A_{ij}}}{\sum_s \pi_s \prod_j \theta_{sj}^{A_{ij}}},$$

$$\pi_r = 1/n * \sum_i q_{ir} \quad \text{and} \quad \theta_{rj} = \frac{\sum_i A_{ij} q_{ir}}{\sum k_i^{\text{out}} q_{ir}}, \qquad (3)$$

Where k_i^{out} is the outdegree of node i. Their algorithm iterates over the three equations to convergence. Thealgorithm works for a directed network and expendable toundirected networks, whereas an extension to weightednetworks is not straightforward. A good feature of this method is that it does not require any preliminary indication of what type of structure to look for; the resulting structure is the most likely classification based on the connectivity patterns of the nodes. Therefore, various types of structures can be detected as community structures i.e., a group of densely connected nodes or as multipartite structures, or even mixed patterns of both. The primary drawback of this method is that the number of communities k must be specified; however, k is usually unknown for real networks.

2) BNEM: a fast community detection algorithm using generative models

This model is similar to the generative model used in Newman and Leicht (2007) to describe how an interaction between two actors in the network, i.e., edges, is generated. This Algorithm uses a generative process to model the interactions between social network's actors [40]. Through unsupervised learning and using expectation maximization, an efficient and fast community detection algorithm based on Bayesian network and expectation maximization(BNEM) has proposed. The process explains why an edge exists between two nodes or how an edge is formed. A edge list is used to representation of a network and directed edge notation. An edge between x and y denoted by $(x \rightarrow y)$will be presented in the list where, x is the source or the initiator of the relationship and y is the target or the receiver. A major advantage of this model over the model used in Newman and Leicht (2007) is that it allows us to deal with directed or undirected networks and can address weighted or unweighted networks.

3) Rare: Rank Removal

This algorithm [13] starts by ranking all vertices according basically to Page Rank. The second step lies in removing highly ranked nodes, those nodes are considered as the cluster cores. An expanding phase is performed by adding each removed nodes to any cluster in a manner to increase the density.

4) Is: Iterative Scan

This approach [13] starts by a seed cluster and then adding or removing vertex at each iteration as the density metric progresses. The algorithms stops if there is no more improvement.

5) K-Clique Percolation

This method [14] is introduced by Palla et al to detect overlapping community. The Clique Percolation procedure can be described as follows: initially, given a network N, identify all cliques of size K. Subsequently, a clique graph is created based on adjacent cliques. If two clique share K-1 nodes they are called adjacent.

6) CONGA Algorithm

This approach [15] is considered as a divisive hierarchical clustering and deals with overlapping community. This algorithm adds the process of node split to GN algorithm, which means node i belongs to k communities at the same time. Indeed, to tolerate belonging to several communities, CONGA relies on splitting a node v into {v1, v2}. The incoming edges to v will be divided between v1, v2 and to link v1 and v2 a virtual edge is established. If the edge betweenness of the virtual edge is greater than any real edge, the node must be split. The complexity is O(e3) where e is the number of edges.

7) COPRA Algorithm

Based on the original label propagation algorithm, COPRA algorithm[15] allows each node to carry multiple labels, and the division of overlapping communities can be achieved by the way of extension tag.

8) Local community detection algorithm based on local degree central nodes

This approach proposed in [16] seeks to discover the key nodes of communities based on node centrality. This measure indicates the importance and the influence of a node in the network. Several measures have been proposed in the literature such as closeness centrality, betweennes centrality, and degree centrality. In this algorithm, the authors use the degree centrality which is defined as the number of edges incident to a node: Cd(Vi) = deg (Vi) Where deg (Vi) is the degree of Vi. In the suggested algorithm, firstly, the local maximal degree node is assigned to the community, and then, adds its adjacent nodes to the community. Subsequently, add to the community the node with the lowest degree, and then expand community along this node, in the case where the node could not belong to the same community with the starting node, the starting node in the local community may not be discovered appropriately with the R method. The node with the lowest degree is not added to the community if it has no common neighbors with the starting node. Two adjacent are in the same community if they have no more common neighbors. Hence, bring together the node which has more common neighbors with the starting node and also with results in the highest increase in R. Where: R is local community measure methods.

C. Overlapping and content-based methods

The content-based methods link users and their posted contents via latent topics. Users interested in the same topic are grouped into a community.

1)Author-Topic (AT)model

This model is proposed by Steyvers et al. [9] and explore the relationships among users, documents, topics, and words. It represents a topic as a multinomial distribution over words and models a user as probability distribution over different topics.

2)Author-Recipient-Topic (ART) model

McCallum et al.[17] presented the author-recipient-topic (ART) model to discover users with similar topic interests, which conditions the topic distribution on the sender–recipient relationships.

3) Community-Author-Recipient-Topic (CART) model

Based on the ART model, Pathak etal. [18] introduced a community-author-recipient-topic (CART) for community extraction from the Eron email corpus, by leveraging both topic and document link information from the social network. Peng et al.[43] proposed a unified user profiling scheme which makes good use of all types of co-occurrence information in the tagging data.

3. Problem definition

In this section, we first explain the multi-relational network and user community discovery problem, then the terminologies which is used in community discovery are defined. finally, we present our framework for discovering user communities from multi-relational networks.

3.1 Multi-relational network

We represent a multi-relational network as a graph G=(V, E), where V is the set of actors in the network, and E is a set of edges indicating the connections among actors in V. The actor set is displayed as V={U, T, R, W}={U1,…,Ui>, <T1,…,Tj>, <R1,R2,…,Rl>, <w1,w2,…,wk>},where Ui represents a user, Tj is a comment that user post it, Rl is reply to user comment and wk represent a word in the vocabulary. E={<U1,U2>, …, <U1,T1>, …,<T1,w1>, …,} indicates the relationship among users, comments, and words. For example, In Twitter, each user is called a twitterer, who can post tweets with a limit of 140 characters, or reply to tweets posted by her friends. Each twitterer could follow any other twitterer she/he is interested in without securing permission. The edge <U1, U2>indicates that twitterer U1 has followed U2.The edge <U1,T1> implies the twitterer U1 has posted tweet T1. The edge <T1,w1> indicates that tweet T1 is composed of word w1. The user structure network associated with G is a subgraph N=<U,Eu>, where Eu⊂E is a set of edges among users. In the twitter case, Eu={<U1, U2>, <U3,U2>,…}. The fact that is a twitterer U1 followsU2 does not necessarily imply that U2 has followed U1.

3.2 User structure network

Considering above scenario, the user structure network is a directed network, and is represented by directed graph N=(U,E_U) that U, E_U are a set of users and relation between them, respectively. For a directed network, the corresponding adjacent matrix A is asymmetric, with Aij=1 indicating user j has marked user i as friend. The network has $|U|$ =n users, and there is $|E_u|$ =m interactions between users of network. A community C in such network is a group of nodes having high density of edges among the nodes and a low density of edges between different groups. Community detection involves identifying the number of k communities or groups in a network and assigning communities membership for each node. Typically, the number of communities k is unknown.

3.3 Document content-based modeling with topics

A number of recent approaches to modeling document content are based upon the idea that the probability distribution over words in a document can be expressed as a mixture of topics, where each topic isa probability distribution over words (e.g., Blei, et al., 2003; Hofmann, 1999). In this section, we will useLatent Dirichlet Allocation (LDA; Blei et al., 2003) topic model to train topic models on Twitter. Latent Dirichlet Allocation topic model is an unsupervised machine learning technique which identifies latent topic information in large document collections. It uses a “bag of words” approach, which treats each document as a vector of word counts. Each document is represented as a probability distribution over some topics, while each topic is represented as a probability distribution over a number of words. In LDA, the generation of a document collection is modeled as a three step process. First, for each document, a distribution over topics is sampled from a Dirichlet distribution. Second, for each word in the document, a single topic is chosen according to this distribution. Finally, each word is sampled from a multinomial distribution over words specific to the sampled topic.

This generative process corresponds to the hierarchical Bayesian model shown in Figure 1. In this model, φ denotes the matrix of topic distributions, with a multinomial distribution over V vocabulary items for each of T topics being drawn independently from a symmetric Dirichlet(β) prior. θ is the matrix of document-specific mixture weights for these T topics, each being drawn independently from a symmetric Dirichlet(α) prior. For each word, z denotes the topic responsible for generating that word, drawn from the θ distribution for that document, and w is the word itself, drawn from the topic distribution φ corresponding to z. Estimating φ and θ provides information about the topics that participate in a corpus and the weights of those topics in each document respectively. A variety of algorithms have been used to estimate the parameters of topic models. we will use Gibbs sampling, as it provides a simple method for obtaining parameter estimates under Dirichlet priors and allows combination of estimates from several local maxima of the posterior distribution.

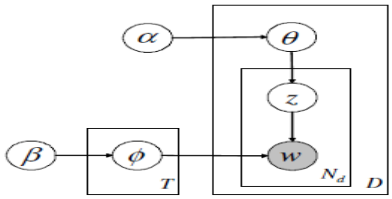

Fig.1. Latent Dirichlet Allocation topic model

The LDA model has two sets of unknown parameters the D document distributions θ, and the T topic distributions φ - as well as the latent variables corresponding to the assignments of individual words to topics z. By applying Gibbs sampling (see Gilks, Richardson, &Spiegelhalter, 1996), we construct a Markov chain that converges to the posterior distribution on z and then use the results to infer θ and φ (Griffiths & Steyvers, 2004). The transition between successive states of the Markov chain results from repeatedly drawing z from its distribution conditioned on all other variables, summing out θ and φ using standard Dirichlet integrals:

$$P(z_i = j | w_i = m, z_{-i}, w_{-i}) \propto$$

$$\frac{C_{mj}^{WT} + \beta}{\sum_{\acute{m}} C_{\acute{m}j}^{WT} + V\beta} \frac{C_{dj}^{DT}}{\sum_{j} C_{dj}^{DT} + T\alpha} \qquad (4)$$

where $z_i = j$ represents the assignments of the ith word in a document to topic j , $w_i = m$ represents the observation that the ith word is the mth word in the lexicon, and z_{-i} represents all topic assignments not including the ith word. Furthermore, C_{mj}^{WT} is the number of times word m is assigned to topic j, not including the current instance, and C_{dj}^{DT} isthe number of times topic j has occurred in document d, not including the current instance. For any sample from this Markov chain, being an assignment of every word to a topic, we can estimate φ and θ using Eq.(5) , Eq.(6):

$$\phi_{mj} = \frac{C_{mj}^{WT} + \beta}{\sum_{\acute{m}} C_{\acute{m}j}^{WT} + V\beta} \qquad (5)$$

$$\theta_{dj} = \frac{C_{dj}^{DT}}{\sum_{j} C_{dj}^{DT} + T\alpha} \qquad (6)$$

where ϕ_{mj} is the probability of using word m in topicj, and θ_{dj} is the probability of topic j in document d.These values correspond to the predictive distributionsover new words w and new topics z conditioned on wand z.

3.4 Learning Automata

A learning automaton [19,20] is an adaptive decision-making unit that improves its performance by learning how to choose the optimal action from a finite set of allowed actions through repeated interactions with a random environment. The action is chosen at random based on a probability distribution kept over the action-set and at each instant the given action serves as the input to the random environment. The environment responds to the taken action in turn with a reinforcement signal. The action probability vector is updated based on the reinforcement feedback from the environment. The objective of a learning automaton is to find the optimal action from the action-set so that the average penalty received from the environment is minimized. The environment can be described by a triple $E \equiv \{\alpha, \beta, c\}$, where $\alpha \equiv \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ represents the finite set of the inputs, $\beta \equiv \{\beta_1, \beta_2, \ldots, \beta_m\}$ denotes the set of the values that can be taken by the reinforcement signal, and $c \equiv \{c_1, c_2, \ldots, c_m\}$denotes the set of the penalty probabilities, where the element c_i is associated with the given action α_i. If the penalty probabilities are constant, the random environment is said to be a stationary random environment, and if they vary with time, the environment is called a non-stationary environment. The environments depending on the nature of the reinforcement signal β can be classified into P-*model*, Q-*model* and S-*model*. The environments in which the reinforcement signal can only take two binary values 0 and 1 are referred to as P-model environments. Another class of the environment allows a finite number of values in the interval [0, 1] can be taken by the reinforcement signal. Such an environment is referred to as a Q-model environment. In S-model environments, the reinforcement signal lies in the interval [*a, b*]. The relationship between the learning automaton and its random environment has been shown in Figure2 [19].

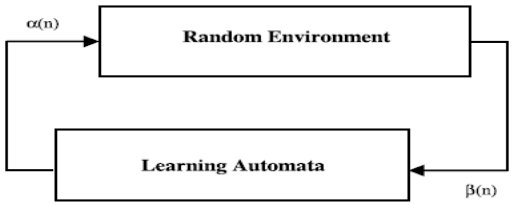

Fig.2. Relation between learning automata and environment

Learning automata can be classified into two main families [19]: fixed structure learning automata and variable structure learning automata. Variable structure learning automata are represented by a triple β, α, *T* , where β is the set of inputs, α is the set of actions, and *T* is the learning algorithm. The learning algorithm is a recurrence relation which is used to modify the action probability vector. Let α(*k*) and *p*(*k*) denote the action chosen at instant *k* and the action probability vector on which the chosen action is based, respectively. The recurrence equation shown by (7) and (8) is a linear learning algorithm by which the action probability vector *p* is updated.

Let α(*k*) be the action chosen by the automaton at instant *k*.

$$p_j(k+1) = \begin{cases} p_j(k) + a[1 - p_j(k)] & j = i \\ (1-a)p_j(k) & \forall j \neq i \end{cases} \qquad (7)$$

when the taken action is rewarded by the environment (i.e. $\beta(n) = 0$) and

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & j = i \\ \left(\frac{b}{r-1}\right) + (1-b)p_j(k) & \forall j \neq i \end{cases} \qquad (8)$$

when the taken action is penalized by the environment (i.e. β(*n*) = 1).

r is the number of actions that can be chosen by the automaton, $a(k)$ and $b(k)$ denote the reward and penalty parameters and determine the amount of increases and decreases of the action probabilities, respectively. If $a(k) = b(k)$, the recurrence equations (1) and (2) are called the linear reward–penalty (L_{R-P}) algorithm, if $a(k) > b(k)$ the given equations are called the linear reward-ε penalty ($L_{R-\varepsilon P}$), and finally if $b(k) = 0$ they are called the linear reward–Inaction (L_{R-I}). In the latter case, the action probability vectors remain unchanged when the taken action is penalized by the environment.

Learning automata have been found to be useful in systems where incomplete information about the environment, wherein the system operates, exists. Learning automata are also proved to perform well in dynamic environments. It has been shown in Ref. [21] that the learning automata are capable of solving the distributed problems. Recently, several learning automata based approaches have been presented for improving the performance of many applications [22-26].

3.4.1 Variable Action Set Learning Automata

A variable action set learning automaton is an automaton in which the number of actions available at each instant changes with time. It has been shown in [39] that a learning automaton with a changing number of actions is absolutely expedient and alsoε-optimal, when the reinforcement scheme is *L-RI*. Such an automaton has a finite set of n actions, $\alpha = \{\alpha_1, \alpha_2, \dots \alpha_n\}$. $A = \{A_1, A_2, \dots A_m\}$denotes the set of action subsets and $A(k) = \alpha$ is the subset of all the actions can be chosen by the learning automaton, at each instant k.

$\hat{p}_i(k) = prob[\alpha(k) = \alpha_i, \quad A(k), \alpha_i \in A(k)]$ is the probability of choosing action α_i, conditioned on the event that the action subset $A(k)$ has already been selected and also $\alpha_i \in A(k)$. The scaled probability$\hat{p}_i(k)$ is defined as:

$$\hat{p}_i(k) = \frac{p_i(k)}{K(k)} \qquad (9)$$

$$K(k) = \sum_{\alpha_i \in A(k)} p_i(k) \qquad (10)$$

$K(k)$, is the sum of the probabilities of the actions in subset $A(k)$, and $p_i(k) = prob[\alpha(k) = \alpha_i]$. The procedure of choosing an action and updating the action probabilities in a variable action set learning automaton can be described as follows. Let $A(k)$ be the action subset selected at instant k . Before choosing an action, the probabilities of all the actions in the selected subset are scaled as defined in equation (3). The automaton then randomly selects one of its possible actions according to the scaled action probability vector $\widehat{p(k)}$. Depending on the response received from the environment, the learning automaton updates its scaled action probability vector. Note that the probability of the available actions is only updated. Finally, the probability vector of the actions of the chosen subset is rescaled as:

$$p_i(k+1) = \hat{p}_i(k+1).K(k) \;\; for\ all\ \alpha_i \in A(k) \qquad (11)$$

The absolute expediency and ε−optimality of the method described above have been proved in [39].

3.4.2 Distributed Learning Automata

A Distributed learning automata (DLA) [27-30] shown in Fig. 2 is a network of interconnected learning automata which collectively cooperate to solve a particular problem. Formally, a DLA can be defined by a quadruple $\{A, E, T, A_0\}$, where $A = \{A_1, A_2, \dots, A_n\}$is the set of learning automata,$E \subset A * A$is the set of the edges in which edge $e_{(i,j)}$ corresponds to the action α_{ij} of the automaton A_i, T is the set of learning schemes with which the learning automata update their action probability vectors, and A_0 is the root automaton of the DLA from which the automaton activation is started.

The operation of a DLA can be described as follows. At first, the root automaton randomly chooses one of its outgoing edges (actions) according to its action probabilities and activates the learning automaton at the other end of the selected edge. The activated automaton also randomly selects an action which results in activation of another automaton. The process of choosing the actions and activating the automata is continued until a leaf automaton (an automaton which interacts with the environment) is reached. The chosen actions, along the path induced by the activated automata between the root and leaf, are applied to the random environment. The environment evaluates the applied actions and emits a reinforcement signal to the DLA. The activated learning automata along the chosen path update their action probability vectors on the basis of the reinforcement signal by using the learning schemes. The paths from the unique root automaton to one of the leaf automata are selected until the probability with which one of the chosen paths is close enough to unity. Each DLA has exactly one root automaton which is always activated, and at least one leaf automaton which is activated probabilistically. For example in Figure3, If automaton A_1 selects α_2 from its action set, then it will be the activated automaton of A_2. Afterwards, the automaton of A_2 will choose one of its possible actions and so on.

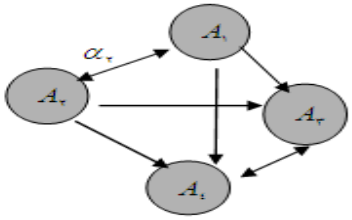

Fig.3. Distributed learning automata

3.5 Proposed Framework

Figure4, shows the overall framework of the our approach. The input is a multi-relational network constructed from social network services. The output is detected community structures and community memberships. Our framework consists of four modules which are Data Generation Module, Content Modeling Module, Learning Module and community detection module. In Data Generation Module, Users–friendship Network extracts from social network analysis and user logfile includes users navigation information from various documents. For Content Modeling Module, topic distribution over documents(θ) and topic distribution over words(φ)are modeled. Learning module consists of Distributed Learning Automata with Changing Number of Actions. The task of the learning module is to learn the relationship between users. In fact, this module by using user-friendship structure and users logfile simultaneously, learns relationships between users. The extracted relationships are used as input of the community detection module and then, one community detection algorithm is applied on them. Finally, the community structures are generated.

The most recent community detection methods just apply structural information of each user in relation to the other users and do not pay attention to content relationship between users in any way. Therefore, considering the interests of the users in visiting various documents in order to discovering similarities between the users is important. Generally, when two users in visiting a document are interested in similar topics, there is more relation or similarity between them. Thus, they must be placed within the same community. The procedure of proposed learning module is as follows: we illustrate user-friendship network as a distributed learning automata with n learning automata which have changing number of actions at any time. There is a learning automata corresponding to any user. Each

learning automata has at most n-1 actions. Set of actions at any instant is consists of the user neighbors at that time. At first, all of the actions are inactive. by visiting a random user of any document on social network, the learning automata corresponding to the user is activated. selected user randomly chooses one of the it's topics in terms of her/his interests. Learning automata related to the selected user randomly selects one action according to its action probabilities from it's active actions set. By choosing this action, the learning automata at the other end of the chosen action is activated. now if, the user corresponding to the activated automata, refers to the same document and topic of previous user, taken action is rewarded, in otherwise remains unchanged. When a learning automata rewards it's action, it should update probability vector corresponding to itself. The process of choosing the actions and activating the automata is continued till all documents and all users are limited to documents are selected. The result of learning module is the users-content network which relations between users with helping distributed learning automata are learned. In continue, we represent how can use the result of learning module in order to discovering user communities.

The applied community detection algorithm has an agglomerative approach and it is built on the notion of edge betweenness [31] that is introduced by Newman and Girvan and also present an improvement of the use of this concept. At first step we need to find initial community core (Coc) in the network. Wediscover the key nodes in the network since those nodes arecharacterized by their influence to other nodes in the graph. To construct the initial partition, we must place each central node in a distinct community. This partition is composed of N communities with N is the number of central node. The second step, we Compute the edge betweenness for each edge in the initial graph. the third step that is related to Community expanding, for each community core, if the adjacent node have a smaller edge betweenness add iteratively the direct neighbor, i.e. adjacent node, add iteratively for each node his direct neighbor [32].If a node has a null betweenness with all central nodes, we put it in the community with which it has a maximum node in common. The last step is Community optimization. In this step two communities are merged if they are highly overlapped community, i.e. they share several nodes.

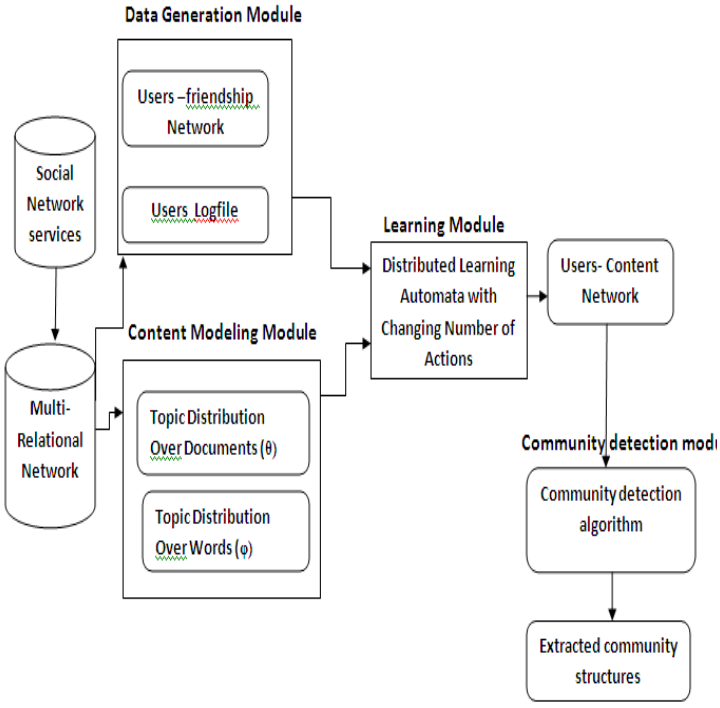

Fig. 4. Our proposed framework

4. Emperical evaluation

4.1 Dataset

In this section, we empirically assess the efficacy of the proposed method using Twitter real world dataset[38]. To illustrate the benefit of our proposed method, pure NMF[35] and NMF-AT methods were used as the benchmark methods to compare against our DLA-AT algorithm. The MetaFac [33] model was also selected as benchmark, which performs tensor factorization on the multi relational network and shares a similar high-level design as ours. In the following subsections, we first introduce evaluation measures. Then, we report the experimental findings on the Twitter dataset, respectively.

4.2 Evaluation Metrics

User and topic are two major components of any characterization of communities. To evaluate the quality of the given communities detected by any algorithms, we use the following user and topic-related measures. The first measure is The mean value $\bar{\mu}$ which is the average value of soft modularity Qs and user-content similarity Su. This measure measures the comprehensive performance of friendship density and content similarity in the extracted communities, of which higher Qs indicating that users in the same community are densely connected with each other, and higher Su indicating that users in the same community share more similar content information with each other. The soft modularity Qs as defined by Newman and Girvan in [34] in order to measure the goodness of a community structure. We represent the content feature of each user as a tf-idf vector over the related words. Given the tf-idf vector representation of each user Vi (i=1,2…m) the users' content similarity value for overlapping community structures is calculated as Eq.(12):

$$S_U = \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{K} R_{ki}R_{kj}\, sim(V_i, V_j)}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\sum_{k=1}^{K} R_{ki}R_{kj}} \qquad (12)$$

Where, $R_{\mathrm{ki}} = 1$ if user i belongs to community k, otherwise, $R_{ki} = 0$; sim(Vi,Vj) indicates the cosine similarity of contents between user i and user j.

Therefore, mean value $\bar{\mu}$ is calculated as follows:

$$\bar{\mu} = \frac{Q_s + S_U}{2} \qquad (13)$$

The next measures are community user divergence$D_U$and community topic divergence D_T to evaluate the diversity ofdetected communities from different perspectives. A higher divergence value generally implies that the communities are better distinguished from each other. We also use a composite divergence measure D, a harmonic mean of D_U and D_T.For a good community partition, the extracted communities should be distinguished from each other. This could be measured by calculating the distance among communities. Jenson–Shannon (JS) divergence has been a popular method for measuring the distance between two probability distributions [36,37]. Based on JS divergence, we define D_U and$D_T$to measure the distance of detected communities from user distribution and community topic distribution, respectively. Given two communities C1 and C2, the community user divergence between C1 and C2 is defined as: $D_U(C1, C2) = JS(\phi_{C1}, \phi_{C2})$, Where$JS(\phi_{C1}, \phi_{C2})$ is the JS divergence [35] between the two probabilistic distributions ϕ_{C1}, ϕ_{C2}. ϕ_{C1}, ϕ_{C2} are user probabilistic distributions over communities C1, C2 respectively. Given two communities C1 and C2, the community topic divergence between C1 and C2 is defined as: $D_T(C1, C2) = JS(\varphi_{C1}, \varphi_{C2})$, where $\varphi_{C1}, \varphi_{C2}$ are topic distributions over words within communities C1, C2. To gain balance between the user and topic divergence in measuring an identified community structure, we calculate the overall community divergence as the harmonic mean of D_U andD_T.Given two communities C1 and C2, the community

divergence betweenC1 and C2 is calculated as: $D(C1,C2)=\frac{D_{U*}D_T}{D_U+D_T}$.
The divergence values are measured pair-wise among communities in the detected community structure S. To avoid clustering, we only report the average divergence value calculated as:

$$\overline{div}=\frac{1}{k(k-1)}\sum_{ci\in s}\sum_{cj\in s,ci\neq cj} div(ci,cj) \qquad (14)$$

Where, div could be D_U, D_T or D.

4.3 Evaluation Results

Figure5, shows the comparison of mean value $\bar{\mu}$among NMF-AT, NMF, MetaFac and DLA-AT models under different community numbers. We observe that DLA-AT improves the performance of community detection by considering the tweets information(content user information). The overall curve trends of the four models maintain the same, they go up to the peak and then drop as k increases. The difference is that DLA-AT performs better than NMF-AT, NMF and Metafac throughout the interval range. Additionally, it can be noted that the best community partition occurs when k=8.The divergence comparison of DLA-AT with benchmark methods on the Twitter dataset is shown in Table1. It can be seen that DLA-AT achieves highest community user divergence value D_U, indicatingour method can group users better into different communities. The MetaFac model performs best in grasping community topics and gains highest community topic divergenceD_T. Also, the DLA-AT model shows Due to the simultaneous use of users friendship information and content-based users relation, in discovering user communities has better performance in comparison to other methods.

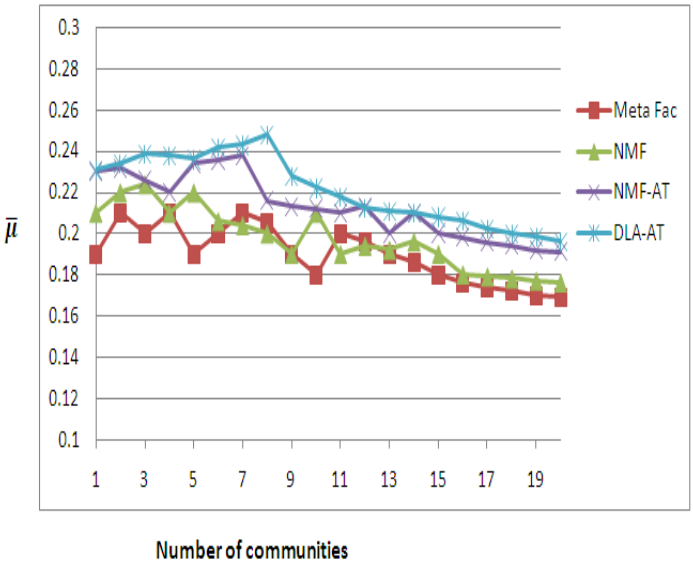

Fig.5. Comparison ofour proposed method(DLA-AT) with previous methods in terms of mean value($\bar{\mu}$) and number of communities.

Table2: Comparison of our proposed method with previous methods in terms of divergence measures.

Twitter Dataset					
	AT	Meta fac	NMF	NMF-AT	DLA-AT
$\overline{D}_U$	0.278	0.259	0.52	0.66	0.73
$\overline{D}_T$	0.583	0.717	-	0.34	0.41
$\overline{D}$	0.329	0.324	-	0.49	0.56

5. Conclusion

In this paper, we proposed a new framework based on distributed learning automata for user community detection on social network. Most recent works has focused on user community detection by analyzing either user friendship networks or user-generated contents but not both at the same time. The advantage of our proposed framework in comparison with other previous works is to consider user-friendship structure and user content information simultaneously. The proposed framework, due to the use of distributed learning automata to learn the content-based relationship between users, plays crucial role and aids in finding significant communities. We design four modules in our framework . By utilizing each module, we are able to improve the modularity properties of discovered community structures. Finally we have evaluated our framework on the twitter dataset. Evaluation results indicate that this framework can discover substantial user communities, which have dense ralationships among community members.

References:

[1]- K. Valck, G.H. Bruggen, B.Wierenga, Virtual communities: a marketing perspective,Decision Support Systems 47 (3) (2009) 185–203.
[2]- T. Spaulding, How can virtual communities create value for business Electronic Commerce Research and Applications 9 (1) (2010) 38–49.
[3]- C. Chiu,M. Hsu, E.Wang, Understanding knowledge sharing in virtual communities: an integration of social capital and social cognitive theories, Decision Support Systems 42 (3) (2006) 1872–1888.
[4]- L. Nie, B. Davison, B. Wu, From whence does your authority come? Utilizing community relevance in ranking, in: The 22nd AAAI Conference on Artificial Intelligence, 2007, pp. 1421–1426.
[5]- F.Y. Wang, D. Zeng, J.A. Hendler, Q. Zhang, Z. Feng, Y. Gao, H. Wang, G. Lai, A study of the human flesh search engine: crowd-powered expansion of online knowledge, Computer (2010).
[6]- M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E 69, 2004.
[7]- G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature, 2005.
[8]- X. He, H. Zha, C.H.Q. Ding, H.D. Simon, Web document clustering using hyperlink structures, Computational Statistics and Data Analysis, 2002.
[9]- M. Steyvers, P. Smyth, M. Rosen-Zvi, T. Griffiths, Probabilistic author-topic models for information discovery, in: The Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 306–315.
[10]- D. Zhou, I. Councill, H. Zha, C. Giles, Discovering temporal communities from social network documents, in: Proceedings of the 7th IEEE International Conference on Data Mining, 2007, pp. 745–750.
[11]- A.K. Jain, M.N. Murty, P.J. Flynn, "Data Clustering: A Review". ACM Computing Surveys," Vol. 31, pp. 264-323, 1999.
[12]- M. E. J. Newman, M. Girvan, "Finding and evaluating community structure in networks," Physical Review, Vol. 70,2004.
[13]- J. Baumes, M. Goldberg, M. Krishnamoorthy, M. MagdonIsmail, N. Preston, "Finding Communities by Clustering a Graph into Overlapping Subgraphs," IADIS International Conference on Applied Computing, 2005.

[14]- L. Tang, H. Liu, "Community Detection and Graph-based Clustering Adapted," 2010.
[15]- S. Gregory, "An Algorithm to Find Overlapping Community Structure in Networks," In Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases, Varsovie, Pologne, pp. 91 – 102, 2007.
[16]- Q. Chen, T.Wu, M. Fang. "Detecting local community structures in complex networks based on local degree central nodes,"Physica A: Statistical Mechanics and its Applications. 2013.
[17]- A. McCallum, X. Wang, A. Corrada-Emmanuelm, Topic and role discovery in social networks with experiments on Enron and academic email, The Journal of Artificial Intelligence Research 30 (2007) 249–272.
[18]- N. Pathak, C. Delong, A. Banerjee, K. Erickson, Social topic models for community extraction, in: The 2nd SNA-KDD Workshop '08, 2008.
[19]- K.S. Narendra, M.A.L. Thathachar, Learning Automata:An Introduction, Prentice-Hall, 1989.
[20]- M.A.L. Thathachar, P.S. Sastry, Networks of Learning Automata: Techniques for Online Stochastic Optimization, Kluwer Academic Publishers, 2004.
[21]- J. Akbari Torkestani, M.R. Meybodi, Finding minimum weight connected dominating set in stochastic graph based on learning automata, Inform. Sci.200 (2012) 57–77.
[22]- A. Rezvanian, M.R. Meybodi, An adaptive mutation operator for artificial immune network using learning automata in dynamic environments, in: Proceedings of the 2010 Second World Congress on Nature and Biologically Inspired Computing, NaBIC, 2010: pp. 479–483.
[23]- A. Rezvanian, M.R. Meybodi, LACAIS: learning automata based cooperative artificial immune system for function optimization, in: Contemporary Computing, Springer, Berlin, Heidelberg, 2010, pp. 64–75.
[24]- A. Rezvanian, M.R. Meybodi, T. Kim, Tracking extrema in dynamic environments using a learning automata-based immune algorithm, in: Grid and Distributed Computing, Control and Automation, Springer, Berlin, Heidelberg, 2010, pp. 216–225.
[25]- J. Akbari Torkestani, A highly reliable and parallelizable data distribution scheme for data grids, Future Gener. Comput. Syst. 29 (2013) 509–519.
[26]- F. Amiri, N. Yazdani, H. Faili, A. Rezvanian, A novel community detection algorithm for privacy preservation in social networks, in: A. Abraham (Ed.), Intelligent Informatics, 2013, pp. 443–450.
[27]- H. Beigy, M.R. Meybodi, Utilizing distributed learning automata to solve stochastic shortest path problems, Int. J. Uncertain. Fuzz. 14 (2006) 591.
[28]- J. Akbari Torkestani, M.R. Meybodi, Clustering the wireless Ad Hoc networks: a distributed learning automata approach, J. Parallel Distrib. Comput. 70 (2010) 394–405.
[29]- R. Forsati, M.R. Meybodi, Effective pagerecommendation algorithms based on distributed learning automata and weighted association rules, Expert Syst. Appl. 37 (2010) 1316–1330.
[30]- M. Soleimani-Pouri, A. Rezvanian, M.R. Meybodi, Solving maximum clique problem in stochastic graphs using learning automata, in: 2012 Fourth International Conference on Computational Aspects of Social Networks, CASoN, 2012, pp. 115–119.
[31]- M.Girvan, M. E. J.Newman. Community structure in social and biological networks. Proceedings of the National Academy of Sciences, volume 99, pp. 7821-7826, 2002.
[32]- Q. Chen, T.Wu, M. Fang. Detecting local community structures in complex networks based on local degree central nodes. Statistical Mechanics and its Applications. 2013.
[33]- Y. Lin, J. Sun, P. Castro,R.Konuru,H. Sundaram,A. Kelliher,MetaFac: community discovery via relational hypergraph factorization, in: The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 527–536.
[34]- M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E 69 (2004) 026113.
[35]- S. Zhang, R. Wang, X. Zhang, Uncovering fuzzy community structure in complex networks, Physical Review E 76 (4) (2007) 046103.
[36]- J. Lin, Divergence measures based on the Shannon Entropy, IEEE Transactions on Information Theory 37 (1) (1991) 145–151.
[37]- M. Steyvers, T. Griffiths, Probabilistic topic models, in: Handbook of Latent Semantic Analysis, 2007.
[38]- M. Choudhury, Y. Lin, H. Sundaram, K. Candan, L. Xie, A. Kelliher, How does the data sampling strategy impact the discovery of information diffusion in social media? in: The 4th Int'l AAAI Conference on Weblogs and Social Media, 2010.
[39]- Thathachar , MAL and Harita, B.: Learning automata with changing number of actions. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 17(6), 1095–1100 (1987).
[40]- A.Ibrahem Hafez, A. Hassanien, BNEM: a fast community detection algorithm using generative, Social networkAnalysis mining journal, springer journal, 29 July 2014.
[41]- Nam P. Nguyen, M. Alim, Thang N. Dinh, A method to detect communities with stability in social networks, Social networkAnalysis mining journal, springer journal, 25 July 2014.

Rahebeh Mojtahedi Saffari received her B.S. degree in Computer Engineering from Islamic Azad University Lahijan branch, Iran in 2004 and M.S. degree in Computer Engineering from Islamic Azad University Qazvin branch, Iran in 2008. Since 2013, she had been pursuing her Ph.D.degree in Computer Engineering at Islamic Azad University Qazvin branch in Iran. Her research focuses on Learning Automata, Social Networks (User Community Detection, User Behavior Modeling),Web Personalization and Web Recommender systems. Her current employment is Associate Professor of Islamic Azad University Lahijan Branch since 2008.

Hassan Rashidi Associate Professor. Allameh Tabataba'i University; Qazvin Islamic University. His research focuses on Software Engineering, Scheduling Algorithms, DSS.