Identification of Web Communities using Cellular Learning Automata

S. Motiee **M. R. Meybodi**
Soft Computing Laboratory
Computer Engineering and Information Technology Department
Amirkabir University of Technology
Tehran Iran
motiee@aut.ac.ir, mmeybodi@aut.ac.ir

Abstract: A collection of web pages which are about a common topic and are created by individuals or any kind of associations that have a common interest on that specific topic is called a web community. Since at present, the size of the web is over 3 billion pages and it is still growing very fast, identification of web communities has become an increasingly hard task. In this paper, a method based on asynchronous cellular learning automata (ACLA) for identification of web communities is proposed. In the proposed method first an asynchronous cellular learning automaton is used to determine the related pages and their relevance degree (the relationship structure of web pages). For determination of relationship structure of web pages information about hyperlinks and the users' behaviour in visiting the web pages are used. Then, an algorithm similar to the HITS algorithm is applied on the obtained structure to identify the web communities. One of the advantages of the proposed method is that the web community obtained using this method is not dependent on a specific web graph structure. To evaluate the proposed approach, it is implemented and the results are compared with the results obtained for two existing methods, HITS and a complete bipartite graph based method. Experimental results show the superiority of the proposed method.

Keywords: Web Mining, Web Community, Cellular Learning Automata, HITS Algorithm, Web Usage Data

1. Introduction

World Wide Web has been growing rapidly in recent years and this has resulted in a huge volume of hyperlinked documents which contain no logical organization. Currently, Google indexes more than 3 billions web pages in the world which this number increases with the rate of 7.3 million pages per day. To utilize this enormous volume of information, web mining techniques have been introduced in recent years. One of the categories of web mining is web structure mining which obtains information about web pages and their relationships through the existing hyperlinks between these pages. In this type of web mining, WWW is modelled by a graph in which the web pages are the graph nodes and the hyperlinks are the graph edges. Web structure mining is used for various purposes such as ranking web pages, identification of web communities, analysis of web graph and simulation of web graph generation process. A web community is a collection of web pages which are about a common topic and are created by individuals or any kind of associations with a common interest on that topic [2]. By identification of web communities users can obtain useful information about that topic through using the pages of web community. Since currently the volume of WWW has exceeded 3 billion web pages and it is increasing continuously, the identification of web communities is getting extensively difficult.

Numerous approaches are introduced for identification of web communities which can be categorized into two classes: hyperlink analysis based approaches and graph theory based approaches. The approaches which are introduced in [2] and [3] are based on hyperlink analysis. The approach presented in [2] receives an initial collection of web pages as input and acquires the web communities which these web pages contain. This method is based on an algorithm named RPA[1] which finds the pages relevant with one page via hyperlink analysis. RPA algorithm is applied on every page in the initial set. Then according to the similarity between the obtained results, the pages are divided in some groups and web communities are acquired. The method presented in [3] which is one of the most important approaches for identification of web communities introduces a collection of hub and authority pages as a web

community. An authority page is a page which contains valuable information about a specific topic and a hub page is a page which contains hyperlinks to pages with relevant information to a specific topic. This method identifies hub and authority pages using HITS algorithm [7].

The approaches which are reported in [4], [5], [6] and [8] are based on graph theory. The graph theory based approaches analyze the web graph; but since the web is so huge and expanding, the graph algorithms can not be used simply. To use these algorithms in the web, they should be resistant against incomplete and unknown data. In graph theory based approaches web communities are defined as dense parts of web graph. However, the structure of the dense sub graph in each approach is different. For example, Kumer and his colleagues in [4] have obtained the communities through identification of complete bipartite graphs in the web. They acquire the web communities during web crawling via a technique named Trawling. In [5] another approach for identification of web communities using complete bipartite graph is introduced. In this approach a collection of web pages are considered as input for the algorithm. At first, all complete bipartite graphs ($K_{3,3}$) are extracted from the vicinity graph of these pages. Then these sub graphs are merged and generate the web communities. Apart from complete bipartite graph based approaches, some other methods have also applied graph theory to identify the web communities. In methods presented in [6] and [8], a web community is recognized as a collection of web pages such that each member page has more hyperlinks with in the community than outside of the community. In these methods, a web community is obtained by separating a sub graph from the web through using maximum flow algorithm.

The approaches that are suggested for web community identification till now only use the hyperlinks between web pages. However, by only using the hyperlinks it is not possible to extract the conceptual relationship between web pages completely. In this paper, an approach based on asynchronous cellular learning automata is suggested to identify web communities. In this approach, apart from the hyperlinks between web pages, the user behaviour in navigating these pages is used for identification of web communities. The proposed approach in this paper has two main steps. In the first step, a method based on asynchronous cellular learning automata which uses hyperlinks between pages and user navigation behaviour for determination of relationship structure of web pages (the related pages and their relevance degree) is proposed, i.e. the related pages and their relevance degree is determined. In the second step, by applying an algorithm similar to the HITS algorithm on the structure obtained in the previous step, the web communities that are related to arbitrary topics are acquired. For evaluation, the proposed approach is implemented and its results are compared to the results of HITS algorithm and a complete bipartite based algorithm [5]. The experimental results show the superiority of the proposed method.

The rest of the paper is organized as follows: In section 2 learning automaton, cellular learning automata and asynchronous cellular learning automata are briefly introduced. Section 3 explains the HITS algorithm. In section 4, the proposed approach and in section 5 the simulation results are presented. The final section is conclusion.

2. Learning Automata, Cellular Automata and Cellular Learning Automata

In this section, cellular automata, learning automaton, cellular learning automata and asynchronous cellular learning automata are briefly introduced.

Cellular Automata: Cellular automata are mathematical models for systems consisting of large number of simple identical components with local interactions. CA is a non-linear dynamical system in which space and time are discrete. It is called cellular because it is made up of cells like points in a lattice or like squares of checker boards, and it is called automata because it follows a simple rule. The simple components act together to produce complicated patterns of behaviour. Cellular automata perform complex computations with a high degree of efficiency and robustness. They are especially suitable for modelling natural systems that can be described as massive collections of simple objects interacting locally with each other [15, 16]. Informally, a d-dimensional CA consists of an infinite d-dimensional lattice of identical cells. Each cell can assume a state from a finite set of states. The cells update their states synchronously on discrete steps according to a local rule. The new state of each cell depends on the previous states of a set of cells, including the cell itself, and constitutes its neighbourhood [17]. The state of all cells in the lattice is described by a configuration. A

configuration can be described as the state of the whole lattice. The rule and the initial configuration of the CA specify the evolution of CA that tells how each configuration is changed in one step.

Learning Automata: Learning Automata are adaptive decision-making devices operating on unknown random environments. The Learning Automaton has a finite set of actions and each action has a certain probability (unknown for the automaton) of getting rewarded by the environment of the automaton. The aim is to learn to choose the optimal action (i.e. the action with the highest probability of being rewarded) through repeated interaction on the system. If the learning algorithm is chosen properly, then the iterative process of interacting on the environment can be made to result in selection of the optimal action. Figure 1 illustrates how a stochastic automaton works in feedback connection with a random environment. Learning Automata can be classified into two main families: fixed structure learning automata and variable structure learning automata (VSLA). In the following the variable structure learning automata is described.

A VSLA is a quintuple $< \alpha, \beta, p, T(\alpha,\beta,p) >$, where α, β, p are an action set with s actions, an environment response set and the probability set p containing s probabilities, each being the probability of performing every action in the current internal automaton state, respectively. The function of T is the reinforcement algorithm, which modifies the action probability vector p with respect to the performed action and received response. Let an ACLA operates in an environment with $\beta=\{0,1\}$. Let $n \in N$ be the set of nonnegative integers. A general linear schema for updating action probabilities can be represented as follows. Let action i be performed then

If $\beta(n)=0$ then

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$

$$p_j(n+1) = (1-a)p_j(n) \qquad \forall j \quad j \neq i$$

If $\beta(n)=1$ then

$$p_i(n+1) = (1-b)p_i(n)$$

$$p_j(n+1) = (b/s-1) + (1-b)p_j(n) \quad \forall j \quad j \neq i$$

Where a and b are reward and penalty parameters. When $a=b$, the automaton is called L_{RP}. If $b=0$ the automaton is called L_{RI} and if $0<b<<a<1$, the automaton is called $L_{R\varepsilon P}$. For more Information about learning automata the reader may refer to [9].

Cellular Learning Automata: Cellular learning automata, which is a combination of cellular automata (CA) and learning automata (LA), is a powerful mathematical model for many decentralized problems and phenomena. The basic idea of CLA, which is a subclass of stochastic CA, is to use learning automata to adjust the state transition probability of stochastic CA. A CLA is a CA in which a learning automaton is assigned to every cell. The learning automaton residing in a particular cell determines its action (state) on the basis of its action probability vector. Like CA, there is a rule that the CLA operates under. The rule of the CLA and the actions selected by the neighbouring LAs of any particular LA determine the reinforcement signal to the LA residing in a cell. The neighbouring LAs of any particular LA constitute the local environment of that cell. The local environment of a cell is nonstationary because the action probability vectors of the neighbouring LAs vary during evolution of the CLA. The basic idea of CLA, which is a subclass of stochastic CA, is to use learning automata to adjust the state transition probability of stochastic CA. A d-dimensional CLA is formally defined below.

A d dimensional cellular learning automata can be defined as $CLA=(Z^d,\varphi,A,N,F)$ where

- Z^d is a lattice of d−tuples of integer numbers. This network can be finite, infinite or semi-finite.
- φ is a finite set of states.
- A is a set of learning automata each of which is assigned to one of the cells of cellular automata.
- $N=\{x_1,\ldots,x_m\}$ is a finite subset of Z^d and is named neighbourhood vector.
- $F: \varphi_m \rightarrow \beta$ is the local rule of CLA in where β is the set of values that the reinforcement signal can take. Function F computes the reinforcement signal for each LA based on the actions selected by the neighboring LA. It computes the new state for each cell from the current states of its neighbors.

The learning automaton residing in a particular cell determines its action (state) on the basis of its action probability vector. Like CA, there is a rule that the CLA operates under. The rule of the CLA and the actions selected by the neighbouring LAs of any particular LA determine the reinforcement signal to the LA residing in a cell. The

neighbouring LAs of any particular LA constitute the local environment of that cell. The local environment of a cell is nonstationary because the action probability vectors of the neighbouring LAs vary during evolution of the CLA. The basic idea of CLA, which is a subclass of stochastic CA, is to use learning automata to adjust the state transition probability of stochastic CA. A CLA is called synchronous if all LAs are activated at the same time in parallel. A CLA is called asynchronous (ACLA) if at a given time only some LAs are activated independently from each other, rather than all together in parallel. The LAs may be activated in either time-driven or step-driven manner. In time-driven ACLA, each cell is assumed to have an internal clock which wakes up the LA associated to that cell while in step-driven ACLA, a cell is selected in fixed or random sequence. A d dimensional step-driven ACLA with n cells is defined as $ACLA=(Z^d,\varphi,A,N,F,\rho)$ where the first 5 elements are defined the same as CLA and ρ is an n-dimensional vector called activation probability vector, where ρ_i is the probability that the LA in cell i *(for i = 1, . . . , n)* to be activated in each stage. Suppose LA A_i with finite action set α_i is associated to cell i *(for i = 1, . . . , n)* of ACLA. Let cardinality of α_i be m_i and the state of ACLA represented by $p = (p^T_1, p^T_2, \ldots, p^T_n)^T$, where $p_i= (p_{i1}, \ldots _{imi})^T$ is the action probability vector of LA *Ai*. The operation of ACLA takes place as the following iterations. At stage k, each LA A_i is activated with probability ρ_i and the activated LAs choose one of their actions. Then rule calculates the reinforcement signal. The actions of neighbour cells of an activated cell are their most recently selected actions. Let α_i and β_i be the action chosen by the activated LA A_i and the reinforcement signal received by LA A_i, respectively. The reinforcement signal is produced by the application of local rule F. The higher value of β_i means that the chosen action of A_i is more rewarded. Finally, activated LAs update their action probability vectors and the process repeats. For more information about CLA the reader may refer to [12][13][14].

3. HITS Algorithm

HITS algorithm [7] receives the topic of the desired web community as input and builds a neighbourhood graph based on the following approach. At first, a collection of relevant pages to the received topic is fetched using a search engine. This set is called root set. Then, the root set is augmented by its neighbours, which are the set of pages that either hyperlink to or are hyperlinked by pages in root set. Since the indegree of nodes (that is, the number of pages hyperlinking to a page in the root set) can be very large, in practice a limited number of these pages is included. This new set is called base set or neighbourhood graph.

Then, HITS algorithm computes two authority and hub scores for each node in the base set iteratively. The nodes with high authority scores are authority pages and the nodes with high hub scores are hub pages. An authority page contains valuable information about a specific topic and a hub page contains hyperlinks to pages including helpful information about a particular subject. This algorithm assumes a page which points to many other pages, is a good hub and a page which many other pages point to it is a good authority. Recursively, a page which points to many good authorities is a better hub and a page which is pointed by many good hubs is a better authority.

The recursive algorithm for computing hub and authority scores is as follows:

1. N is the set of nodes in base set.
2. For each node A in N, authority score is shown by *Aut[A]* and hub score is shown by *Hub[A]*.
3. The initial value of *Hub[A]* is set one for all the nodes.
4. While the *Aut* and *Hub* scores are not converged:
   a. For all A in N:
   $$Aut[A] = \sum_{(B,A)\in N} Hub[B] \quad (3)$$
   b. For all A in N:
   $$Hub[A] = \sum_{(A,B)\in N} Aut[B] \quad (4)$$
   c. *Hub* and *Aut* vectors are normalized.

After computing the hub and the authority scores for the pages in the base set, a collection of pages with the highest hub and authority scores are introduced as the members of a web community.

4. The Proposed Approach

Existing methods use web graph to identify web communities [2][3][4][5]. However, in the web graph, the relation between pages is determined according to the existing hyperlinks between pages which do not show the conceptual relationship between pages correctly. In the proposed method the relationship structure of web pages is obtained using both hyperlinks and the user navigation behaviour. The relationship structure shows the relevant pages and their relevance degree. Then the

web communities are identified by applying HITS algorithm on the obtained relationship structure. Therefore, the proposed method is composed of two main phases:

Phase 1: Determination of the relationship structure of pages.

Phase 2: Identification of web communities by applying HITS algorithm on the relationship structure. obtained in the first phase.

In the remaining part of this section what the above two phases are described.

4.1. Phase 1: Relationship Structure Determination

An asynchronous cellular learning automaton is used to find the relationship structure of web pages. The cellular space of ACLA is a 2-dimensional array $G=[0,w(n)-1]\times[0,h(n)-1]$ where $h(n)$ and $w(n)$ are set to $2\lfloor\sqrt{n}\rfloor$ and n is the number of web pages. Also, it is assumed that the cellular space's upper bound is connected to its lower bound and the left bound is connected to the right bound. We represent each web page by an agent and equip each agent with a learning automaton. The task of learning automaton is to guide its agent to reach an appropriate cell. Each learning automaton has 8 actions each of which corresponds to moving to one of its 8 neighbouring cells in 2-dimensional ACLA space.

At the beginning of Relationship Structure Determination phase (RSD Phase), the agents are distributed among the cells of ACLA at random. Some cells may not be assigned any agent and remains empty. Then the agents start moving around in cellular space of ACLA according to a moving strategy (described latter) until every agent finds a cell whose neighbours contain agents with the most accordance with it. Activation of a cell of ACLA which contains an agent causes the agent moves to one of the neighbouring cells to that cell according to the moving strategy. Cell selection for activation in ACLA is done according to row major method. Once the last cell (the last cell in the last row) in ACLA is selected a new round of cell section will begin. At each iteration of RSD phase, one cell of ACLA is selected and then activated if the activation value (computed by equation 5) of its agent is greater then a pre-specified threshold.

$$p_a(agent_i)=\frac{\beta^2}{\beta^2+f(agent_i)^2} \quad (5)$$

In equation (5), β is a constant and f computed according to equation (6) is the fitness measure which shows to what extent an agent is related to its neighbours.

$$f(agent_i)=\max\{0,\frac{1}{9}\sum_{agent_j\in N(agent_i)}(1-\frac{d(agent_i,agent_j)}{k})\} \quad (6)$$

In equation (6), $N(agent_i)$ is the set of $agent_i$'s neighbours, k is a constant and $d(agent_i,agent_j)$ is the distance between two web pages corresponding to $agent_i$ and $agent_j$ and is computed according to equation 7.

$$d(agent_i,agent_j)=\sqrt{(s_{i,1}-s_{j,1})^2+\ldots+(s_{i,k}-s_{j,k})^2} \quad (7)$$

In equation (7), $s_{m,n}$ is the relevance degree of page m with n^{th} topic. k is the number of topics in the web. (Calculation of $s_{m,n}$ is explained in section 5)

After a cell is activated, its corresponding learning automaton chooses one of its actions (one of its 8 neighbours) and then the corresponding agent moves to the cell corresponding to that action provided that this cell has not been occupied by any other agent. Let $cell_i$ be the cell that the agent moves into it. If the web page corresponding to $cell_i$ and at least one of the web pages of $cell_i$ neighbours are appeared in a cycle(s) travelled by a web user(s) then the chosen action will be penalized (because the user has been wandering in the web aimlessly returning to a previously visited web page). Otherwise the chosen action is rewarded if one the following cases hold.

- If the web page corresponding to $cell_i$ points to or is pointed by at least one of the web pages of $cell_i$'s neighbours.
- If the web page corresponding to $cell_i$ and at least one of the web pages of $cell_i$'s neighbours have been appeared in a path(s) followed by a web user(s).

The action probability vector of the learning automaton for agent $agent_i$ is updated according to $L_{R\varepsilon P}$ learning algorithm with time varying learning parameters a and b defined according to equations (8) and (9).

$$a=c_1\frac{\sum_{\forall g\in N(agent_i)\,and\,(g\to agent_i\,or\,agent_i\to g)}1}{\sum_{\forall g\in N(agent_i)}1}+c_2\sum_{\forall path_k|agent_i\,and\,N(agent_i)\in path_k}\frac{1}{Length(path_k)} \quad (8)$$

and

$$b = \frac{\sum_{\forall cycle_i | agent_i \text{ and } N(agent_i) \in cycle_i} Length\ (cycle_i)}{\sum_{\forall path_i | agent_i \text{ and } N(agent_i) \in path_i} Length\ (path_i)} \qquad (9)$$

In the above equations $g \rightarrow agent_i$ shows that the web page corresponding to agent g points to web page corresponding to $agent_i$, $Length(path_i)$ is the length of $path_i$ in the web traversed by one user. In the first component of equation 8, the numerator is the number of neighbours of $agent_i$ whose corresponding pages point to or are pointed by the web page corresponding to $agent_i$. The denominator is the number of neighbours of $agent_i$. The reward parameter a is increased if the number of hyperlinks between $agent_i$'s web page and neighbours' web pages is increased. The second component of equation 8 paths which contain web pages which correspond to $agent_i$ and its neighbours are only taken into consideration. If there is no such path, the second component is set to zero. c_1 and c_2 are constants which determine the importance of each of the two above components in the calculation of a parameter. In equation 9, the numerator is the sum of the lengths of the cycles and the denominator is the sum of the lengths of the paths which are navigated by the web users. Only cycles and paths that contain web pages which correspond to $agent_i$ and its neighbours are only taken into consideration.

Algorithm terminates after a pre determined number of rounds which at that time the location of each agent is finalized. At the end of the algorithm the final neighbours of each agent are determined using which the relevance degree for each agent and its neighbours are computed according to equation 10:

$$r(i,j) = f(agent_i)\frac{1}{d(agent_i, agent_j)} \qquad (10)$$

where $r(i,j)$ is the relevance degree of $agent_i$ and $agent_j$ and consequently page i and page j.

Using relevance degrees computed for each agent and its neighbours, the relationship structure of web pages are determined.

The RSD algorithm is described in more details in the algorithm given below.

```
//Relationship Structure Determination Algorithm
Define a two dimensional ACLA and initialize parameters
for each web page do
        assign an agent to a web page
        place agent randomly at cell
        equip agent with LA
end for
while (not termination) // such as iterations exceeds a threshold
        user_log: Array of [Number of Users][Users Path]
                        /* user log, pages viewed by each
                user. Each row contains path of a user. */
        for each cell do // traverse cells by row major method
                compute activation value for agent placed in current
cell using equation (5)
                if (agent's activation value > R) then
                        select one of the actions of agent's LA
randomly
                        move to an unoccupied neighbour cell
                        based on selected action
                        compute reward parameter by equation (8)
                        compute penalty parameter by equation (9)
                        if (penalty parameter!= 0) then
                                penalize(action) // Penalizing the
                        action is done by equation (2)
                        else if (reward parameter!= 0) then
                                reward(action) // Rewarding the
                        action is done by equation (1)
                        end if
                end if
        end for
end while
Relationship Structure Model := compute relevance degree of each
agent and its neighbour using equation (10)
```

4.2. Phase 2: Web Community Identification

The relationship structure obtained in the first phase is used in phase 2 to determine the web community for a specific topic. The algorithm for phase 2 described below is similar to HITS algorithm. The algorithm for phase 2 with its details is given below and the differences between HITS and our algorithm are mentioned afterwards.

1- Creation of Root Set: Using the desired topic (input to the algorithm) a collection of pages that are relevant to this topic are selected and the root set is created. In the proposed algorithm this collection is chosen randomly.

2- Creation of Base Set: The base set is created from the root set by adding the pages that the root set pages are connected (according to the relationship structure produced in phase 1) to the root set.

3- Hub and Authority Scores Calculation: Initially, both hub and authority scores for each page in the base set is set to one. Then the following loop is performed.

```
while ( the difference between the hub and authority scores for each page > 0.2) do
  begin
```

For all $page_i$ in the *BestSet*:

$$Authority\ (page_i) = \sum_{\forall page_j \rightarrow page_i} r(page_j, page_i) \times Hub(page_j) \quad (11)$$

For all $page_i$ in *BestSet*:

$$Hub(page_j) = \sum_{\forall page_i \rightarrow page_j} r(page_i, page_j) \times Authority(page_j) \quad (12)$$

Hub and Authority scores are normalized.

end.

In equation (11) and (12) $r(page_i,page_j)$ is the relevance degree of $page_i$ and $page_j$ in the relationship structure produced in phase 1 and $page_i \rightarrow page_j$ means that $page_i$ points to $page_j$ in this structure.

4 - Web Community Construction: The web community is constructed by selecting 2p pages: *p* pages with the highest hub scores and *p* pages with the highest authority scores where p is constant.

Remark 1: Two approaches are reported in the literature for construction of Web Community (step 4). In the first approach [18] hub pages are not considered as the member of web community. The justification is that the hub pages usually do not contain relevant information about web community and they only have hyperlinks to authority pages. The other reason for excluding hub pages from the web community is mixed hub phenomena which means hub pages may point to various pages which are relevant to different subjects. In the second approach [3] the hub pages are included in the web community. The reason is that the authority pages are connected to each other through hub pages. In the proposed algorithm, we follow the second approach; however, to avoid the mixed hub phenomena, the hub pages that point to more than 3 different subjects are excluded from the web community.

Remark 2: The differences between the algorithm of phase 2 and HITS algorithm are as follows:

- In HITS the base set is created according to web graph; however, in the algorithm of phase 2, the base set is created according to the relationship structure produced in phase 1.
- In the algorithm of phase 2, to calculate the hub and authority scores a coefficient is incorporated into the equations used by HITS algorithm. This coefficient is the relevance degree between $page_i$ and $page_j$ and is computed based on usage data and hyperlinks.

5. Experimental Results

We evaluate the proposed algorithm using two sets of experiments. For the first set of experiments we have used a simulation model which represents web pages and web users and for the second set of experiments data from a real web site has been used.

5.1. Evaluation of the proposed algorithm using the Simulation Model

In [11], Lui and his colleagues have modelled the regularities of users behaviour in the web by an agent based model and validated their model by usage data of several large sites such as Microsoft. In [11], this model is used instead of using real web pages and real web usage data. This model provides an environment containing web pages and users. The advantage of using this model is that identification of users and their page visits are more precise and no data analysis is required. However, the parameters which are introduced in this model should be set carefully so that the result will be similar to real environment. In this model, every web page has a vector. Each item of this vector shows the relevance degree of this web page with the equivalent topic to this item. (The number of topics is constant and can be defined.) The relevance degree is declared with a number between zero and one. The summation of all relevance degrees for a web page (vector's items) is equal to one. (These relevance degrees are used in equation 7 to define the relevance degree of page *m* with n^{th} topic). Moreover, every page has some hyperlinks with other pages. In the experiments, the users' interests' profiles are based on power law distribution and web pages contents are based on normal distribution. The other parameters which are used in this model and their values in our experiments are shown in Table(1).

Table(1). Parameters of Simulation Model

Degree of Coupling	0.7
Number of Users	5000
Number of Web Pages	2000
Number of Topics	10
T_c : Initial Constant Value of a Web Page for Different Topics	0.2

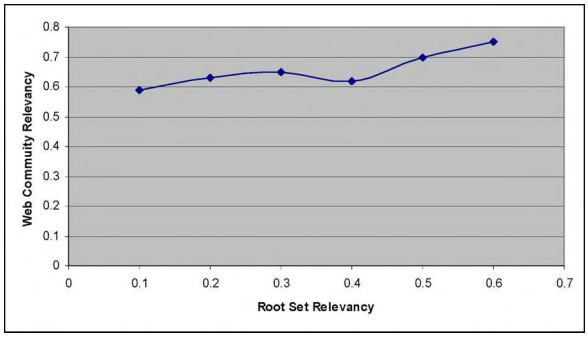
α_u: parameter of a Power Law distribution in Users Preference Distribution	1
λ : Information Absorbing factor of a User from a Web Page	0.5
σ_m: Variance of log-normal Distribution of ΔM^t_t	0.25
μ_m: Mean of log-normal Distribution of ΔM^t_t	5.97
α_p: Shape Parameter of a Power Law Distribution	3
σ_t: variance of normally distributed offset T	0.25
θ: weights of motivation	1

For evaluation of proposed approach, several experiments are done that study the effects of features of proposed approach on relevancy of identified web community. The evaluation measure is the average of relevance degree of web community members with the topic specified by user. The results of these experiments are shown in Figures (1) and (2). Moreover, the efficiency of our approach is compared to two other methods. The first method is HITS algorithm and the second approach is introduced in [5] and is based on complete bipartite graphs. The results of these comparisons are shown in Figures (3) and (4). The values of parameters used for simulation are given in Table (2).

Table(2). Parameters of Proposed Approach

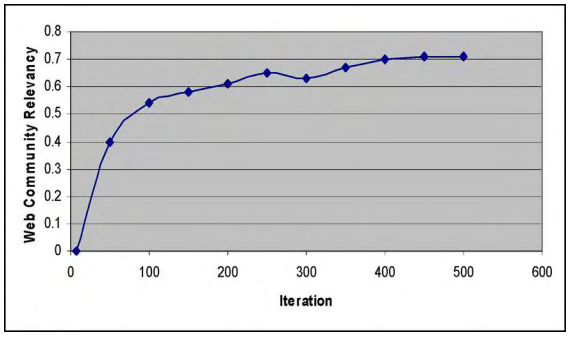
c_1: Importance Quotient of hyperlinks in Reward Calculation	0.05
c_2: Importance Quotient of paths in Reward Calculation	0.05
k: Constant in Calculation of f function	0.5
B: Constant used in Activation Value Calculation	0.1
R: Agent Activation Threshold	0.02
Root Set's Members Count	100
Base Set's Members Count	300
p: Web Community's Members Count	10

Experiment 1: In this experiment, we have studied the effect of root set selection on the relevance of identified web community. For this purpose the average relevance degree of root set pages with declared topic is changed and the average relevance degree of identified web community pages is measured. As it is shown in Figure (1), the relevance degree of web community is almost independent of root set selection.



Figure(1). The effect of Root Set Selection on Web Community Identification

Experiment 2: In this experiment, we have studied the effect of number of iterations which are used in the structure determination algorithm on the quality of identified web community. In each iteration of algorithm, the paths, which are followed by users, are processed and the relevant pages and their relevance degree are updated. As it is shown in Figure (2), when the number of iterations increases, the relevance degree of identified web community increases until further iterations does not have any effect on the results.



Figure(2). The effect of Algorithm's Iteration on Web Community Relevancy

Experiment 3: In this experiment, the proposed approach is compared to HITS algorithm. For this purpose, web communities for 5 different topics are obtained by using our approach and HITS algorithm. This experiment is repeated 10 times and the average of results is used. As it is shown in Figure (3), the relevance degree of the web communities which are obtained by our approach is more than HITS algorithm. The efficiency of HITS algorithm depends on the quality of base set pages (The number of pages that are relevant to the topic). In expansion of root set to base set, usually a considerable number of irrelevant pages are added to base set. Therefore, in many cases HITS algorithm consider irrelevant pages as the members of web community. However, since in the proposed approach, hub and authority scores are

calculated based on the user navigation behaviour, the number of irrelevant pages is decreased.

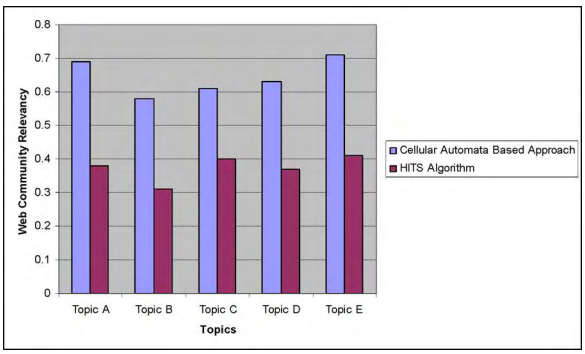


Figure (3). The comparison of proposed approach with HITS algorithm

Experiment 4: In this experiment, the proposed approach is compared to a complete bipartite graph based algorithm which is introduced in [5]. This approach is an example of graph based approaches. As it is shown in Figure (4), the relevance degree of members of identified web community in proposed approach is more than this algorithm. One of the reasons of this increase is that the web communities which are identified by our approach are not dependent on a specific graph structure, however, the approach which is introduced in [5], searches one specific graph structure (bipartite graph) in the web. Since the structure of web communities are not limited to one or some specific structures, this algorithm can not work efficiently. Moreover, in the algorithm presented in [5], if the density of hyperlinks between relevant pages is more, the quality of identified web community will be more as well. But since in our approach we have used user navigation in addition to hyperlinks, effect of the aforementioned factor has decreased.

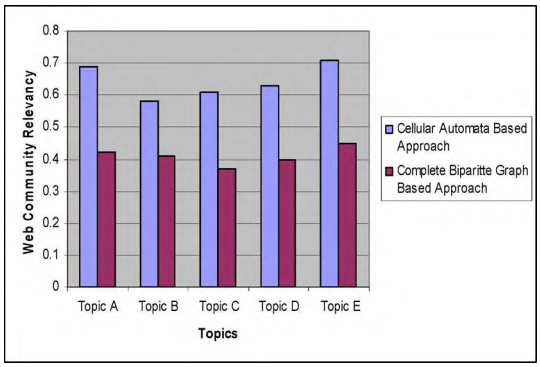


Figure (4). The comparison of proposed approach with graph based algorithm

5.2. Evaluation the proposed algorithm using Real Web Data

This data set contains the pre processed and filtered sessionized data for the main DePaul CTI Web server (http://www.cs.depaul.edu). The data is based on a random sample of users visiting this site for a 2 week period during April of 2002. The original (unfiltered) data contained a total of 20950 sessions from 5446 users. The filtered data files were produced by filtering low support page views, and eliminating sessions of size 1. The filtered data contains 13745 sessions and 683 page views. We have used the unfiltered data because the repetitive page views are not eliminated so that we can study the effect of cycles on algorithm performance. To use this data set, the subjects of web site are extracted which are as follows: admission, advising, authenticate, course, graduate application, news, people, programs, research, scholarship. Since in addition to users' paths, the hyperlinks between web pages are used in the algorithm, these hyperlinks are extracted. Moreover, the file containing the page views is processed in a way that in each step a subset of paths is used. The parameters which are used for performing experiments on real data are mentioned in Table3. In order to compare results for real data with the results for simulation model, the simulation model experiments are repeated and the parameters are set according to Table3 except the hyperlinks which are about 9412 in simulation model.

Table(3). Parameters of Real Web Data Set

Number of Users	683
Number of Web Pages	5446
Number of Topics	10
Number of Hyperlinks	5027
Number of Navigated Paths	20952

The evaluation measure is the average of relevance degree of web community members with the topic specified by user. Since the real web pages are not available (The data set is gathered in 2002), a number showing the web page relevancy with web community topic (between 0 and 1) is assigned to each member according to keywords in its URL. In the following the results of two experiments are explained which show the efficiency of the proposed algorithm on real web data.

Experiment 5: In this experiment, the web communities for 5 topics are obtained by applying proposed algorithm on simulation model and real web data. Each experiment is repeated 5 times and the average result is presented. It should be noted that simulation model topics are different from real

web data topics. As it is shown in Figure (5), the relevancy of web community obtained from simulation model is more than the relevancy of web community obtained from real web data. The reason is that the number of hyperlinks in simulation model is more than the number of hyperlinks in real web data. Moreover, in simulation model, all the hyperlinks are created based on relevancy between pages. However, such an assumption is not always valid for real web pages.

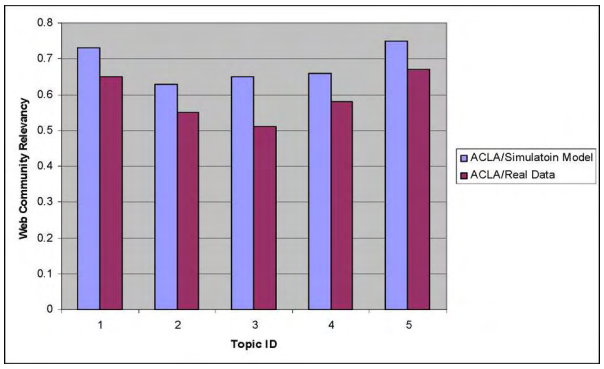


Figure (5). The comparison of proposed real web data with simulation model

Experiment 6: In this experiment, the web communities obtained by applying proposed algorithm on real web data are compared to web communities obtained by applying HITS and complete bipartite graph based algorithms [5] on real data. As it is shown in Figure (6), the web communities' relevancy of proposed algorithm is more than the web communities' relevancy of the other two algorithms. However, the amount of increase in relevancy is less than anount of increase when simulation model is used.

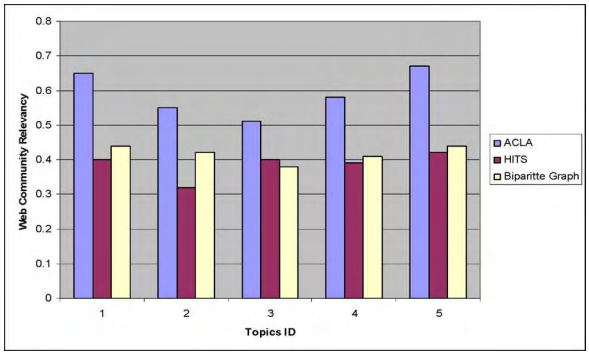


Figure (5). The comparison of proposed real web data with simulation model

From the results of these experiments we can conclude that user navigation behaviour represents the conceptual relation between pages and applying usage data can improve the performance of web community identification algorithms.

6. Conclusion

In this paper, a new approach for identification of web communities was proposed. For this purpose, first an asynchronous cellular learning automaton based algorithm finds the related pages and their relevance degree (the relationship structure of web pages) using web usage data and hyperlinks. Then, an algorithm similar to HITS algorithm identifies the web community related to an arbitrary topic using obtained relationship structure. The results of comparison of our approach to two other algorithms show that applying web usage data can improve the performance of web community identification algorithm because these data convey extra information about web pages relationship. Moreover, the web communities which are identified using our approach are not dependent on any specific web graph structure and they do not suffer from mixedhub phenomena.

References

[1] Beigy, H. and Meybodi, M. R, "*A Mathematical Framework for Cellular Learning Automata*", Advances on Complex Systems, Vol. 7, Nos. 3-4, pp. 295-320, 2004.

[2] Toyoda, M., Kitsuregawa, M., "*Creating a Web Community Chart for Navigating Related Communities*", In Proc. Hypertext 2001, pp.103-112, 2001.

[3] Gibson, D., Kleinberg, J. M., Raghavan, P., "*Inferring Web Communities from Link Topology*", In Proc. of the 9th ACM Conference on Hypertext and Hypermedia. Pittsburgh, PA, pp. 225-234, 1998.

[4] Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A., "*Trawling the Web for Emerging Cyber-Communities*", Proc. of the 8th WWW Conference, 1999.

[5] Imafuji, N., Kitsuregawa, M., "*Effects of Maximum Flow Algorithm on Identifying Web Community*", Proc. of the 4th international Workshop on Web information and Data Management (McLean, Virginia, USA, November 08 - 08, 2002). WIDM '02. ACM Press, New York, NY, pp. 43-48, 2002.

[6] Flake, G., Lawrence, S., Giles, C.L., "*Efficient Identification of Web Communities*", the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston, MA, pp. 150-160, 2000.

[7] Kleinberg, J., "*Authoritative Sources in a Hyper-linked Environment*", Proc. Of ACM-SIAM Symposium on Discrete Algorithms, 1998. Also appears as IBM Research Report RJ 10076(91892) May 1997.

[8] Flake, G. W., Lawrence, S., Giles, C. L., Coetzee, F. M., "*Self-Organization and Identification of Web Communities*", IEEE Computer, Vol. 35, No. 3, pp. 66-71, 2002.

[9] Narendra, K. S. and Thathachar, M. A. L., *Learning Automata: An Introduction*, Prentice Hall, 1989.

[10] Chen, X. Xu, and Chen, Y., "*A Novel Ant Clustering Algorithm Based on Cellular Automata*", Proc. IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 04), 2004.

[11] Liu, J., Zhang, S. and Yang, J., "*Characterizing Web Usage Regularities with Information Foraging Agents*," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 4, pp. 566-584, 2004.

[12] Beigy, H. and Meybodi, M. R., "Open Synchronous Cellular Learning Automata", Advances in Complex Systems, Vol. 10, No. 4, pp. 1-30, December 2007.

[13] Beigy, H. and Meybodi, M. R., "Asynchronous Cellular Learning Automata", Automatica, Journal of International Federation of Automatic Control, Vol. 44, No. 5, pp. 1350-1357, May 2008.

[14] Beigy, H. and Meybodi, M. R., "A Mathematical Framework for Cellular Learning Automata", Advances on Complex Systems, Vol. 7, Nos. 3-4, pp. 295-320, September/December 2004.

[15] M. Mitchell, "Computation in cellular automata: A selected review", *Technical report,* Santa Fe Institute, Santa Fe, NM, USA,
September 1996.

[16] N. H. Packard, S. Wolfram, "Two-dimensional cellular automata", *Journal of Stat. Phys. 38*, pp. 901-946, 1985.

[17] J. Kari, "Reversibility of 2D cellular automata is undecidable", Physica D45, pp. 379-385, 1990.

[18] Zhou, W., Wen J., Ma, W., Zhang, H., "A Concentric-Circle Model for Community Mining", *Microsoft Technical Report*
MSR-TR-2002-123, Nov. 2002.