Clustering-based web page prediction

Ruma Dutta*

Netaji Subhash Engineering College,
West Bengal University of Technology,
Garia, Kolkata, 700152, West Benagal, India
E-mail: rumadutta2006@gmail.com
*Corresponding author

Anirban Kundu

Kuang-Chi Institute of Advanced Technology,
Software Building, No. 9 Gaoxin Zhong,
1st Road, High-Tech Industrial Estate,
Nanshan District, Shenzhen,
Guangdong, 518057, China
E-mail: anik76in@gmail.com

Debajyoti Mukhopadhyay

Maharashtra Institute of Technology,
S. No. 124, Paud Road, Kothrud,
Pune 411038, Maharashtra, India
and
Web Intelligence and Distributed Computing Research Lab
(WIDiCoReL), GreenTower, C-9/1,
Golf Green, Kolkata, 700095, India
E-mail: debajyoti.mukhopadhyay@gmail.com

Abstract: Web page prediction plays an important role by predicting and fetching probable web page of next request in advance, resulting in reducing the user latency. The users surf the internet either by entering URL or search for some topic or through link of same topic. For searching and for link prediction, clustering plays an important role. Besides the topic, navigational behaviour is not ignored. This paper proposes a web page prediction model giving significant importance to the user's interest using the clustering technique and the navigational behaviour of the user through Markov model. The clustering technique is used for the accumulation of the similar web pages. Similar web pages of same type reside in the same cluster, the cluster containing web pages have the similarity with respect to topic of the session. The clustering algorithms considered are K-means and K-mediods, where K is determined by HITS algorithm. Finally, the predicted web pages are stored in form of cellular automata to make the system more memory efficient.

Keywords: web page prediction; HITS algorithm; clustering; cellular automata; CA.

Reference to this paper should be made as follows: Dutta, R., Kundu, A. and Mukhopadhyay, D. (2011) 'Clustering-based web page prediction', *Int. J. Knowledge and Web Intelligence*, Vol. 2, No. 4, pp.257–271.

Biographical notes: Ruma Dutta was an Assistant Professor of Netaji Subhash Engineering College (India). She received her BE in Electrical Engineering and ME in Computer Science and Technology from Bengal Engineering College (BESU). She received her PhD (Engg) from Jadavpur University. She had worked for organisations like TCS, PwC, etc. Her research interest is in web mining, cellular automata, data mining and distributed databases.

Anirban Kundu is currently a Research Fellow of Kuang-Chi Institute of Advanced Technology. He received his PhD (Engg) from Jadavpur University. He was an Assistant Professor and the Head of the Department of IT. His research interests include search engine-oriented indexing, ranking, prediction and web page classification with essence of cellular automata. He is also interested in multi-agent-based system design, fuzzy controlled systems and cloud computing. He has authored 35 journals and conference papers.

Debajyoti Mukhopadhyay is the Dean (R&D) of MIT Group of Institutions and the Head of IT at Maharashtra Institute of Technology Pune, India. He is the Founder Director of the Web Intelligence and Distributed Computing Research Lab. He holds Adjunct Professorship at UGSM-Monarch Business School, Switzerland, and Thapar University, India. He had worked at Bell Communications Research, USA in its Computing Systems and Architecture Lab (1987–1994). He received his BE (Electronics) from Calcutta University (India), DCS (Computer Science and Applications) from The Queen's University of Belfast (UK), MS (Computer Science) from Stevens Institute of Technology (USA), and PhD (Computer Science) from Jadavpur University (India).

1 Introduction

The World Wide Web (WWW) has created an opportunity to disseminate and gather information online from vast database of information. This motivates the researchers to study web usage behaviour of the web users by reducing the access latency using efficient web prediction technique. Researchers use different techniques which usually employs Markov model of order-k for next web page prediction in real time (Pitkow and Pirolli, 1999; Kroeger et al., 1997). Navigational behaviour of the user is being recorded in web-log as an input for Markov model. The main limitation of Markov model is as follows: lower order Markov models are coupled with low accuracy; whereas, higher order Markov models are associated with high state-space complexity; and also the sequences are not available in the web-log. This problem motivates us to integrate some other features to be considered with lower order Markov model for better prediction accuracy. Moreover, when a web page comes first time, then most similar web page to current web page can be predicted through clusters which are based on contents and link structure. The logic behind it, web surfers either enter URL or search topic or go by navigation through link. While users go by URL, Markov model is the best option and for other two cases clustering plays a key role.

There are three categories of prediction approaches of web page pre-fetching. These are as follow:

1 Client-based – This pre-fetching model is based on navigation behaviour of a specific server which is stored in cache at client's side. This model is aimed for a particular user or group of users (where web pages are accessed through proxy server).

2 Server-based – This model is based on navigation behaviour of a specific server for the web pages hosted on that server. In this case, different clients use the same server and same website. Priority may be given to some clients, which should be considered while predicting web pages for that server.

3 Client-server-based – As the name suggests this model is the combination of the above two models.

The proposed approach is client-based. It aims at the navigational behaviour of clients as well as client interest.

The novelty of this technique is it can be used as customised web page ranking searching a topic. That means when the user searches a topic, if that particular or similar type of topic has already been searched by the user, the web pages according to usage behaviour of user and of the same topic can be pre-fetched for the user.

The contributions of this paper are

- it integrates Markov model and clustering technique to predict web page
- in clustering technique, for partitional clustering like K-means and K-mediod, the HITS algorithm used to determine the no. of clusters of the web pages in the web-log and the initial centroids of the clusters
- cellular automata is used to reduce memory storage of prediction results of the web pages of websites.

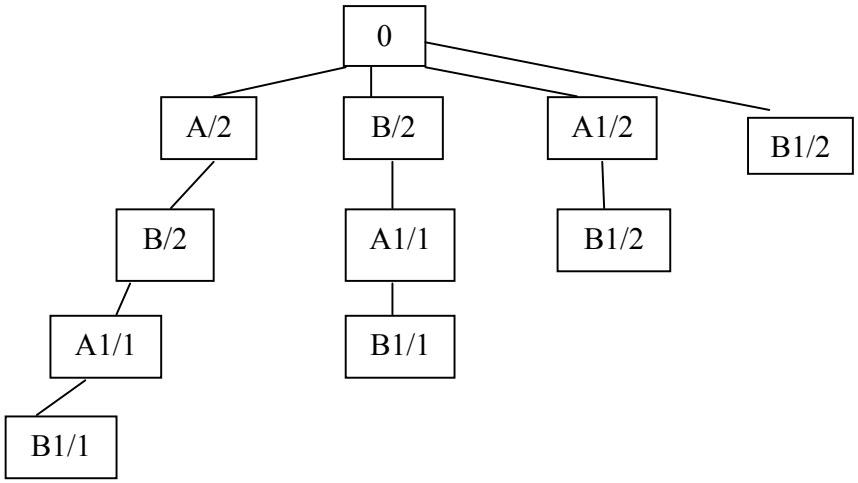
This paper is organised as follows: related works are discussed in Section 2. Section 3 discusses about cellular automata preliminaries. Section 4 discusses about proposed model. Section 5 shows the experimental result and Section 6 concludes the paper.

2 Related works

There are several architectures and related algorithms for developing web predictor. Most of the researchers emphasised on the Markov model and N-grams. An order-k Markov predictor is a scheme which calculates the conditional probability 'p' of accessing web page 'P', such that previous accesses were in order of P_1, P_2, …, P_k. More formally, the existing predictor models mostly follow the equation the probability p = Probability ($P \mid P_k P_{k-1}, \ldots, P_1$) for computing purpose. N^{th}-order Markov models, when parameterised by a length of N, essentially represent the same functional structure as N-grams. These systems analyse the past access history on the web server, maps the sequential access information in N consecutive cell (known as N-grams) for building

prediction models. N-gram methods include two sub-methods: 'point-based' and 'path-based'. 'Point-based' prediction makes the prediction using the last visited URL, which is precisely 1st order Markov model does. In contrast, 'path-based' prediction model uses more than one web page as the observation in order to make prediction. 'Path-based' prediction basically resembles with higher order Markov model. As it is already discussed, 1st order Markov model or 'point-based' prediction does not make accurate prediction since it neglects the previously visited web page information to discriminate the different access patterns. As a result, 'path-based prediction' is more popular. In this area, there is a question on how to choose the best N for N-grams. Bernardo et al. (1998), Davison (2004), Duchamp (1999), Pitkow and Pirolli (1999), Palpanas and Mendelzon (1999), Xin Chen and Zhang (2003), and Su et al. (2000) discussed several different ways to build N-grams and empirically compared their performance on real web-log data. An empirical study was performed (Su et al., 2000) on the tradeoffs between precision and applicability of different N-gram models, showing that the longer N-gram models make predictions accurately with a sacrifice on coverage. There was another approach suggested a way to make predictions based on K^{th}-order Markov models (Pitkow and Pirolli, 1999). Since they prefer longer paths more than shorter ones, their algorithm has the shortcoming that the longer paths are rarer in the web-log history, thus the noise in longer paths could be higher than in shorter paths. This may result in reduction in prediction accuracy. Prediction by partial match (PPM) (Palpanas and Mendelzon, 1999) is a very popular model in the area of web page prediction. This model considers variable order Markov model and the most appropriate order for the prediction of next web page can be chosen. The sample model is given in Figure 1 of this paper. But this type of model requires huge memory space. This problem can be solved by using cellular automata (Dutta et al., 2007). The disadvantage of any Markov model is that it solely depends on the history. This paper solves this problem by considering other web pages of the same cluster. The association rule is also popular in determining the next web pages. The problem with this method is, it considers only frequent sets.

Figure 1 PPM Model for sessions (ABA1B1) (AB) (A1B1)

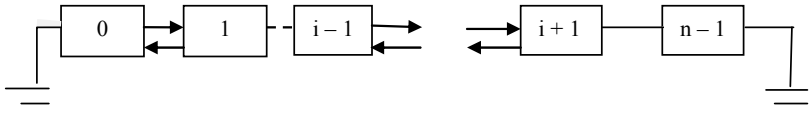


Clustering is a challenging topic in web data management too. Web data clustering is the process of grouping web data into 'clusters' such that similar objects are in the same class and dissimilar objects are in different classes. Its goal is to organise data circulated over the web into groups/collections in order to facilitate data availability and accessing, and at the same time meet user preferences. The problem with any type of clustering is that the number of clusters has to be determined apriori. This problem is solved with the HITS algorithm by the clustering techniques used in this paper. Main benefits of web page clustering include: increasing web information accessibility, understanding users' navigation behaviour, improving information retrieval and content delivery on the web.

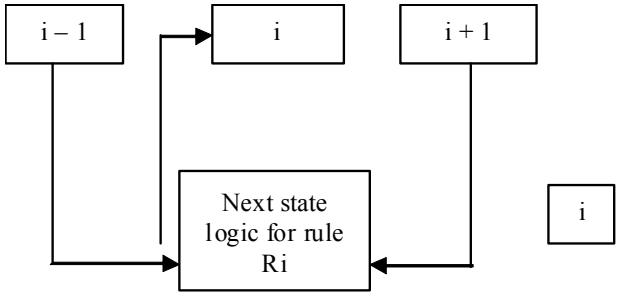
3 Cellular automata preliminaries

In this section, a brief discussion on what is meant by deterministic cellular automata is given. An n cell cellular automata consists of n cells with local interactions. It evolves in discrete time and space [Figure 2(a)]. The next state function of three neighbourhood CA cell [Figure 2(b)] can be represented as a rule as defined in Table 1 (Chaudhuri et al., 1997; Wolfram, 1986). First column of Table 1 represents $2^3 = 8$ possible present states of 3 neighbours of i^{th} cell – $(i - 1)$, i, $(i + 1)$ cells.

Figure 2 Local interactions between cellular automata cells, (a) an *n* cell CA with null boundary (b) the i^{th} cell configured with rule Ri



(a)



(b)

Each of the eight entries (three bit binary string) represents a minterm of a three variable Boolean function for a three neighbourhood CA cell.

In subsequent discussions, each of the eight entries in Table 1 is referred to as a rule min term (RMT). The decimal equivalent of eight minterms are 0, 1, 2, 3, 4, 5, 6, 7 noted within () below the three bit string. Each of the next five rows of Table 1 shows the next

state (0 or 1) of i^{th} cell. Hence, there can be $2^8 = 256$ possible bit strings. The decimal counterpart of such an 8 bit combination is referred to as a CA rule (Chaudhuri et al., 1997). The rule of a CA cell represents its next state logic as illustrated in Table 2 for a few example rules. It can be derived from the truth table (Table 1) of the i^{th} cell, where q_i^{t+1} is the next state of i^{th} cell, while q_{i-1}^t, q_i^t and q_i^{t+1} are the current states of $(i-1)^{th}$, i^{th}, and $(i+1)^{th}$ cells respectively; the ($\oplus$) represents XOR logic and ($\cdot$) denotes AND function.

If a CA cell is configured with a specific rule, its next state function implements the truth table as illustrated for sample rules in Table 2. The first two rules 90 and 150 of Table 1 are linear rules employing XOR logic while remaining non-linear rules employ AND logic in addition to XOR. Out of 256 possible rules, there are seven rules with XOR logic and another seven rules employ XNOR logic. The Rule 0 sets the cell to state '0' for each of the eight minterms. The remaining rules are non-linear rules employing AND/OR/NOT logic. Linear and additive CA employing XOR/XNOR logic have been characterised with matrix algebraic formulation (Chaudhuri et al., 1997). The typical state transition behaviour is given in Figure 3.

Figure 3 State transition behaviour of cellular automata for rule vector <120 90 60 240>

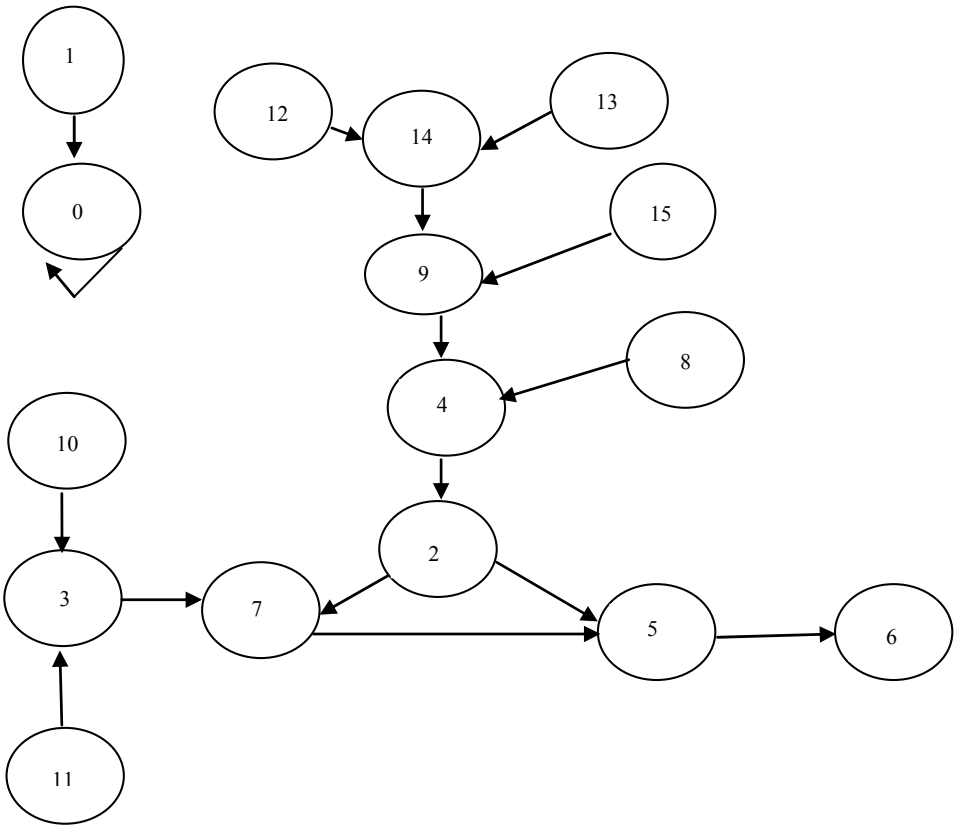


Table 1 Truth table of sample rules of a CA cell showing the next state logic for the minterms of a three variable Boolean function

Present states of 3-neighbours ($i-1$), i, and ($i+1$) of i^{th} cells (Minterms of a three variable Boolean function)	111 (7) T(7)	110 (6) T(6)	101 (5) T(5)	100 (4) T(4)	011 (3) T(3)	010 (2) T(2)	001 (1) T(1)	000 (0) T(0)	Rule numbers
Next state of i^{th} cell	0	1	0	1	1	0	1	0	90
	1	0	0	1	0	1	1	0	150
	0	1	1	1	1	0	0	0	120
	0	0	0	0	1	1	0	0	12

Notes: The eight minterms having decimal values 0, 1, 2, 3, 4, 5, 6, 7 are referred to as rule minterms (RMTs). Set of minterms $T = \{7, 6, 5, 4, 3, 2, 1, 0\}$ represented as $\{T(7), T(6), T(5), T(4), T(3), T(2), T(1), T(0)\}$ ($T(m) = m$, $m = 0$ to 7) in the text, are noted simply as q.

Table 2 Linear/additive CA rule

With XOR (linear CA)	*With XNOR (complemented rule)*
Rule 60: $q_i(t+1) = q_{i-1}(t) \oplus q_i(t)$	Rule 195 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_i(t)}$
Rule 90: $q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$	Rule 165 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_{i+1}(t)}$
Rule 150: $q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$	Rule 153 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)}$

3.1 Definitions

Definition 3.1 (Uniform CA): If all the CA cells obey the same rule, then the CA is said to be uniform CA.

Definition 3.2 (Hybrid CA): If the CA cells obey different rules, then the CA is said to be hybrid CA.

Definition 3.3 (Null boundary CA): A CA is said to be a null boundary CA (NBCA) if the left (right) neighbour of the leftmost (rightmost) terminal cell is connected to logic 0-state [Figure 1.1(a)].

Definition 3.4 (Reachable state): A state having 1 or more predecessors is a reachable state.

Definition 3.5 (Non-reachable state): A state having no predecessor is termed as non-reachable. In Figure 2, 1, 2, 13, 6, 10, 11, 15, 8 are non-reachable states.

Definition 3.6 (Cyclic state): A state in a cycle of the state transition behaviour of a CA is known as cyclic state. Cyclic states are also referred to as stable (semi stable) states.

Definition 3.7 (Transient state): A non-cyclic state of a CA is referred to as a transient state.

4 Proposed model

Definition 4.1 (Session): Let L be a web log. A session S is ordered list of web pages accessed by a user, i.e., $S = <<p_1, t_1>, <p_2, t_2>, …., <p_n, t_n>>$, where there is a user $u_i \in U$ such that $\{<u_i, p_1, t_1>, <u_i, p_2, t_2>, …, <u_i, p_n, t_n>\} \subseteq L$. Here $t_i <= t_j$ iif $i <= j$.

Definition 4.2 (Transition probability 1st order): The transition probability is the probability of going to one of the next state (i.e., next web page) from current state (i.e., web page). For 1st order Markov model, current web page decides the next web page. The formula for transition probability for 'n' number of web pages is given in equation (1).

$$TP_i^j = \frac{AC_i^j}{\sum_{k=1}^{n} AC_i^k} \tag{1}$$

where

i current web page

j next web page in the web log

AC_i^j number of accesses from i^{th} web page to j^{th} web page

TP_i^j transition probability from i^{th} web page to j^{th} web page.

The explanation of calculating transition probability is given later where actual process is explained.

Definition 4.3 (Cosine similarity): Cosine similarity is the most popular distance measure between two document vectors. The formula of cosine similarity between two document vectors is given in (2)

$$\cos\left(\vec{d1}, \vec{d2}\right) = \frac{dot\left(\vec{d1}, \vec{d2}\right)}{\|\vec{d1}\|\|\vec{d2}\|} \tag{2}$$

where

$$dot\left(\vec{d1}, \vec{d2}\right) = d1[0] * d2[0] + d1[1] * d2[1] + ...$$
$$\|\vec{d1}\| = sqrt\left(d1[0] * d1[0] + d1[1] * d1[1] + ...\right)$$
$$\|\vec{d2}\| = sqrt\left(d2[0] * d2[0] + d2[1] * d2[1] + ...\right)$$

Example 1: Let us consider the documents $d1$ and $d2$ with vectors of $d1$ is (1, 2, 1, 2, 1, 1, 0) and (1, 0, 1, 2, 1, 1, 1) respectively. The $dot(d1, d2)$ is 1*1 + 2*0 + 1*1 + 2*2 + 1*1 + 1*1 + 1*1 = 9

$$|| \text{d1} ||= \text{sqrt}\ (1*1 + 2*2 + 1*1 + 2*2 + 1*1 + 1*1 + 0*0) = 3.464$$

$$||\text{d2}\ ||= \text{sqrt}(1*1 + 0*0 + 1*1 + 2*2 + 1*1 + 1*1 + 1*1) = 3$$

Cosine similarity = 9 / (3.46 * 3) = .866

As discussed previously, our prediction technique in this paper employs clustering technique.K-means and K-mediod clustering used in this paper utilise cosine similarity the distance measure. Now, the number of clusters assumed is itself an area of problem as it has been observed that the performance of cluster depends on number of clusters and determining initial centroids. We have applied HITS algorithm (Nomura et al., 2004) to find number of clusters. According to well known Kleinberg's hyperlinked induced topic search (HITS) algorithm, two types of web pages are used in WWW: authorities and hub web pages. The web pages present in the authority websites are known as authority web pages. Authorities are web pages that are recognised as providing significant, trustworthy, and useful information on a topic. The web pages present in the hub websites are known as hub web pages. Hubs are indexing web pages that provide lots of useful links to relevant content web pages. The authority score of a web page is proportional to the sum of hub scores of web pages linking to it, and conversely, its hub score is proportional to the authority scores of the web pages to which it links. In matrix notation, this translates to the following pair of equations:

$$\vec{a} = E^T \vec{h} \tag{3}$$

$$\vec{h} = E\vec{a} \tag{4}$$

where

$\vec{a}$ is the authority score of a web page

$\vec{h}$ is the hub score of a web page

E is the edge set of the web.

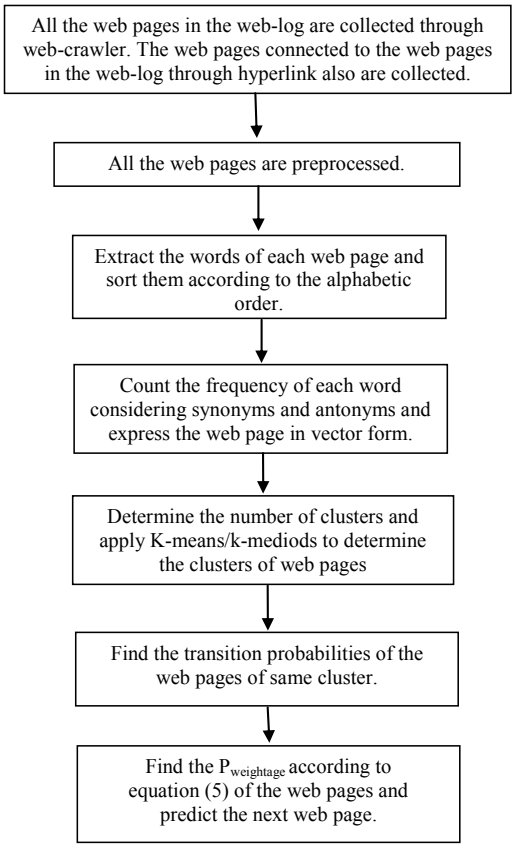
Here, we consider the content of the documents of the 'authorities' web page and that of web pages of a specific server for which cluster has to be decided. The web pages which have the higher authority score more than that of the average of the graph are considered as centroids. Like there are three web pages in the graph p_1, p_2, p_3 and their authority scores have been calculated by equation (3) as $a(p_1)$, $a(p_2)$, $a(p_3)$. Then their average authority score is a $(a_v) = (a(p_1) + a(p_2) + a(p_3)) / 3$. If $a(p_1) > a(a_v)$ and $a(p_2) > a(a_v)$ and $a(p_3) < a(a_v)$ then there are two clusters. So, by the property of hyperlink structure of the web, we have been able to determine the number of clusters. Once number of clusters determined, the web-log records are grouped into clusters. K-means and K-medoid have been used in this paper as clustering techniques. In determining the clusters, the steps in Figure 4 have been followed.

Steps of the proposed approach in this paper are given in Figure 4.

Explanation of Figure 4 is given below and is formulated in Algorithm 1 and Algorithm 2.

The web pages are first preprocessed. In preprocessing all the stop words like 'the', 'and', 'is', 'are' are eliminated. Stemming, i.e., removing prefix like 'ing', 'es', 's' from tokens, is done next (Miller et al., 1993). For efficient searching, words of documents are sorted.

Figure 4 The steps for predicting web page proposed by this paper



To find out transition probability, the equation (1) has been used. The transition probability is actually the probability of occurrence a particular web page from the current web page. Suppose, there are web pages *W*1, *W*2, *W*3 in web-log and the web-log contains the record *W*1 –> *W*2 –> *W*1 –> *W*3. So, the transition probability of *W*2 page from *W*1 is .5 and the web page of *W*3 web page from *W*1 web page is .5. Now, the web pages are to be represented in vector form. To do that, suppose there are three words in *W*1 such as {cat, dog, and mouse}. In *W*2 there are three words {cat, dog, and cow} and in *W*3 there are words {cat, mouse}. The representative vectors are *W*1 = {1, 0, 1, 1}, *W*2 = {1, 1, 1, 0} and *W*3 = {1, 0, 0, 1}. The words of similar meaning (synonyms) as one word and every occurrence of synonymous word increases the frequency of word by 1. Every occurrence of opposite meaning (antonym) deceases the frequency of word by 1. So, if *W*1 contains cat, kitten, dog, mouse words the *W*1 can be expressed by (2, 0, 1, 1). Then, K-means and K-medoid can be applied on these vectors and cluster them in K groups. Suppose the centroids, found from HITS algorithm is *W*1 and *W*2. Dutta et al. (2009) discussed the method of finding the number of clusters and the centroids. For calculating the cosine similarity between the web pages, Definition 3 has to be referred. The cosine similarity of *W*1 and *W*3 is 0.05. The cosine similarity of *W*2, *W*3 is .333. So, the clusters are {*W*1}, {*W*2 and *W*3}.

The model that has been evolved is given in equation (5).

$$P_{weightage} = w1 * Sim\ (wp1,\ wp2) + w2 * Tran\_prob(wp2) \quad (5)$$

where

$wp1$ current web page

$wp2$ candidate web page

$Sim(wp1, wp2)$ cosine similarity between $wp1$ and $wp2$

$Tran\_prob\ (wp2)$ transition probability from $wp1$ to $wp2$

$w1$ and $w2$ are adjustable parameters which have been found out to be .5 through experiments.

The algorithm for finding clusters is given below. The algorithm uses cosine similarity function as cosine similarity is most popular measure in case of textual data.

Algorithm 1 Find clusters

Input:	Web log records $R = \{r_1, r_2, \ldots, r_n\}$
Output:	set of clusters $C = \{c_1, c_2, \ldots, ck\}$
Step 1:	All the web pages in R are collected. The child web pages also are collected in a set S.
Step 2:	Loop (until end of S)
Step 3:	Take a web page from R
Step 4:	Extract the words of the web page
Step 5:	Find out the frequency of each word considering synonyms and antonyms of each word.
Step 6:	Represent them in vector form using cosine similarity.
Step 7:	Loop end
Step 8:	Apply K-means or K-mediod
Step 9:	Find the cluster set C.
Step 10:	Stop.

The total algorithm to find predicted next web page is given in Algorithm 2.

Algorithm 2 Find_Predicted_Next_Web_Page

Input:	set of clusters ($C = \{c_1, c_2, \ldots, c_k\}$), current web page wp
Output:	predicted next web page
/*When a web page is requested, the prediction engine follows the following steps to find the predicted web page.*/	
Step 1:	Find the cluster to which current web page belongs.
Step 2:	If it is the same session and session time < Time threshold then
Step 3:	Find out the $P_{weightage}$ value defined in equation (5) for all the web pages within the Cluster to which current web page belong.
Step 4:	Identify the web page with maximum value of $P_{weightage}$ as predicted web page.
Step 5:	Else
Step 6:	Identify the candidate web page with maximum value of transition probability as predicted web page.
Step 7:	End If
Step 8:	End

Let there be a web-log with sequence ABACAD in a session. It can be observed from this record that if A is the current URL, the candidates web pages are B, C, D. If A, B, C, D are in the same cluster, the system calculates the $P_{weightage}$ for B, C, D and consider, the $P_{weightage}$ value for B is highest, and then B is the predicted web page. If B is not in the same cluster with A, then the $P_{weightage}$ value will be calculated for C and D. If the highest value is for D, then D will be predicted web page. For another case, Time in a session > Time threshold or different session has been started. The logic behind considering the session or time is generally it is common behaviour of human beings to spend time on a similar topic for certain time, then switch to some other topic. In that case, for the log sequence ABACACAD, the candidate web pages are B, C, D. then as transition probability of C is greater, C is the predicted web page.

Moreover, when the user is online if preference differs from the predicted web pages, and if it happens 50% time for some web page then this actual web page replaces the predicted web page in the next session.

Furthermore, for a new web page if it does not fall under any predefined class, the predicted next page will be the hyperlink which has the highest number of in-link.

Algorithm 3 CA rule synthesis

Input:	State transition diagram
Output:	CA rule
Step 1:	Take one non-reachable state from state transition diagram.
Step 2:	Generate key value for the state.
Step 3:	Take next state from that state in state transition diagram.
Step 4:	Generate key value for the state taken in Step 3
Step 5:	Put the value in RMT
Step 6:	If conflict occurs go to Step 4
Step 7:	Else go to Step 3.
Step 8:	Repeat Step 1 to Step 7 until all the states are visited.
Step 9:	Store CA rule.
Step 10:	Stop.

Lastly, the prediction result can be stored in form of cellular automata rule where each web page is a state of cellular automata in state transition behaviour. As an example, please refer to the state transition behaviour in Figure 3, where current web page is a state and predicted page is next web page. This state transition behaviour is stored as CA rule vector <120 90 60 240>. By using the Algorithm 3, cellular automata rule can be synthesised from state transition behaviour of web-log records. By generating the CA rule, the memory requirement for storing the web pages and their predicted web pages is reduced. The detailed analysis of memory requirement for maintaining link list structure and storing CA rule is given in Dutta et al. (2006).

Let us suppose there are three web pages, *W*0, *W*1, and *W*2. The next web page of *W*0 is *W*1, next web page of *W*1 is *W*2 and next web page of *W*2 is *W*0. This can be realised through the state transition diagram. The state transition diagram is 0 –> 1 –> 2 –> 0 where *W*0 is 0, *W*1 is 1, *W*2 is 2. The rule is <2, 1>. Here, we have 3 states, so 2 bytes are required as through 2 bytes we can have $2^2 = 4$ states. Now the states are 00, 01, and 10.

For the state 00, we have two RMTs 000 and 000; for 01 we have RMTs 001, 010; for 10 RMTs are 010 and 100.

The RMT table for the above case is depicted in Table 3.

Table 3 RMT table

111	110	101	100	011	010	001	000	Rule
×	×	×	×	×	0	1	0	2
×	×	×	0	×	0	×	1	1

5 Experimental result

Experiments are carried out with 2 datasets for websites http://www.iihm.org.in and http://www.ihrmcal.org.in. The records are preprocessed, for example, for same web pages appearing consequently, one web page is discarded from web-log. The experiments have been carried out in 128 MB RAM, 20 GB hard disk, and AMD Semprom (TM) processor, 1.40 GHz machine.

Following table (Table 4) shows the result of applying K-means algorithm on http://www.iihm.org.in website. This web site contains 28 web pages. The HITS algorithm found there are 11 clusters. The centroids and the web pages of each cluster are shown in each row of the table.

The quality of clusters after applying K-means and K-mediod algorithm is shown in (Dutta et al., 2009).

After considering both transition probability of usage of a particular client and similarity function value following results are found which is listed in Table 5.

The web page 2, falls into the cluster 3 as shown in Table 4. Cluster 3 contains 13, 3, 4, 5, 7, 9, 21, 24, 25, and 27. web pages along with web page 2. The web-log data contains 3, 13, 4, 5, 7, 9, and 21 web pages at one instance after web page 2. The $P_{weightage}$ for each of these web pages are shown in Table 5. According to the Algorithm 2, for web page 2, predicted next web page is found to be 3.

Table 4 The clusters resulted after implementing K-means algorithm

Centroids	*Other web pages*
11	
12	6, 8
13	2, 3, 4, 5, 7, 9, 21, 24, 25, 27
14	
15	
16	
17	10
18	1
19	23, 26, 28
20	
22	

Table 5 Experimental result for web page 2 in K-means clustering

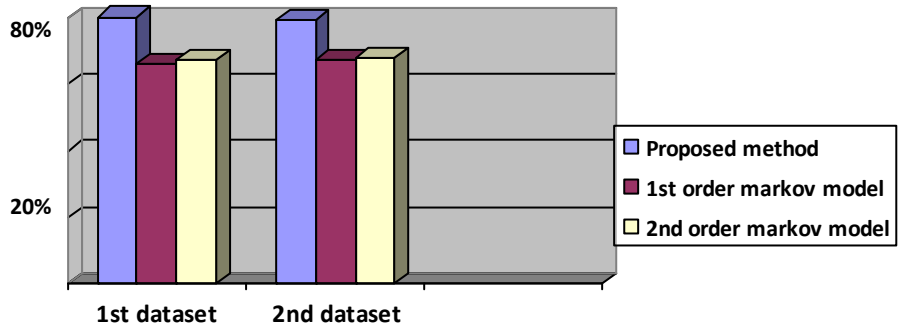
Current web page	*Next web page*	$P_{weightage}$
2	3	.35
2	13	..075
2	4	.15
2	5	.10
2	7	.233
2	9	.017
2	21	.06

So, for web page 2, predicted next web page is 3.

For calculating prediction accuracy, the equation (6) has been followed.

$$\text{Prediction accuracy} = \frac{\text{Number of correct predictions}}{\text{Number of total predictions}} \tag{6}$$

Figure 5 Prediction accuracy (see online version for colours)



Experiments have been carried out for second order Markov model and prediction by partial match (Palpanas and Medelzen, 1999). The first one shows low coverage and second one shows lower prediction accuracy as most of the cases it reduces to 1st order Markov model.

6 Conclusions

The presented approach considers both transition probability and similarity function value in predicting web pages. This approach is particularly suitable for the case of client-based prediction system where searching is used. Apart from the above fact, it is concluded that when higher order Markov model is not suitable for low coverage, this method gives an acceptable result. For a new web page, this prediction technique can predict the next web page as it does not solely depend on history.

The work of this paper can be improved by representing document in TF-IDF vector-space model. The prediction also can be improved by considering time spent on each document by the user, recorded in the web-log of the user. This work is left to the future.

References

Bernardo, A., Huberman et al. (1998) 'Strong regularities in World Wide Web surfing', *Science*, April, Vol. 280, No. 5360, pp.95–97.

Chaudhuri, P.P., Chowdhury, D.R., Nandi, S. and Chatterjee S. (1997) 'Additive cellular automata, theory and applications, Vol. 1', IEEE Computer Society Press, Los Alamitos, California, ISBN-0-8186-7717-1.

Davison, B.D. (2004) 'Learning web request patterns', *Web Dynamics – Adapting to Change in Content, Size, Topology and Use*, pp.435–459, Springer.

Duchamp, D. (1999) 'Pre-fetching hyperlinks', *USENIX Symposium on Internet Technologies and Systems*.

Dutta, R., Ghosh, I., Kundu, A. and Mukhopadhayay, D. (2009) 'An advanced partitioning approach of web page clustering utilizing content & link structure', *Journal of Convergence Information Technology*, Vol. 4, No. 3, pp.65–71, Republic of Korea.

Dutta, R., Kundu, A. and Mukhopadhyay D. (2006) 'An alternate approach for efficient web page prediction', *International Conference on Electronics & Information Technology Convergence*, pp.197–203.

Dutta, R., Kundu, A. and Mukhopadhyay, D. (2007) 'Offering memory efficiency utilizing cellular automata for Markov tree based web-page prediction model', *International Conference on Information Technology*, pp.252–257, India.

Kroeger, T.M., Long, D.D.E. and Mogul, J.C. (1997) 'Exploring the bounds of web latency reduction from caching and pre-fetching', *USENIX Symposium on Internet Technologies and Systems*.

Miller, G., Beckwith, R., FellBaum C., Gross D., Miller, K. and Tengi, R. (1993) *Five Papers on Word Net*, August, Princeton University, USA.

Nomura, S., Oyama, S., Harames, T. and Ishida, T. (2004) 'Analysis and improvement of HITS algorithm for detecting web communities', *Systems and Computers in Japan*, Vol. 35, No. 13, pp.32–42.

Palpanas, T. and Mendelzon, A. (1999) 'Web pre-fetching using partial match prediction', *Appeared in Web Caching Workshop*, March, San Diego, CA, USA.

Pitkow, J.E. and Pirolli, P. (1999) 'Mining longest repeating subsequences to predict World Wide Web surfing', *USENIX Symposium on Internet Technologies and Systems*, pp.139–150.

Su, Z., Yang, Q., Lu, Y. and Zhang, H. (2000) 'Whatnext: a prediction system for web requests using n-gram sequence models', *First International Conferences on Web Information Systems and Engineering Conferences*, Hong Kong, June, pp.200–207.

Wolfram, S. (1986) 'Theory and application of cellular automata', *World Scientific*.

Xin Chen, X. and Zhang, X. (2003) 'A popularity-based prediction model for web pre-fetching', IEEE Computer Society, March, pp.63–70.