DOI: 10.21917/ijsc.2010.0011

TDCCREC: AN EFFICIENT AND SCALABLE WEB-BASED RECOMMENDATION SYSTEM

K.Latha[1], P.Ramya[2], V.Sita[3] and R.Rajaram[4]

Department of Computer Science and Engineering, Anna University, Tiruchirappalli, India
E-mail: erklatha@gmail.com[1], rramyya@yahoomail.com[2], sita_v@ymail.com[3]
[4]Department of Information Technology, Thiagarajar College of Engineering, Madurai, India
E-mail: rrajaram@tce.edu

Abstract

Web browsers are provided with complex information space where the volume of information available to them is huge. There comes the Recommender system which effectively recommends web pages that are related to the current webpage, to provide the user with further customized reading material. To enhance the performance of the recommender systems, we include an elegant proposed web based recommendation system; Truth Discovery based Content and Collaborative RECommender (TDCCREC) which is capable of addressing scalability. Existing approaches such as Learning automata deals with usage and navigational patterns of users. On the other hand, Weighted Association Rule is applied for recommending web pages by assigning weights to each page in all the transactions. Both of them have their own disadvantages. The websites recommended by the search engines have no guarantee for information correctness and often delivers conflicting information. To solve them, content based filtering and collaborative filtering techniques are introduced for recommending web pages to the active user along with the trustworthiness of the website and confidence of facts which outperforms the existing methods. Our results show how the proposed recommender system performs better in predicting the next request of web users.

Keywords:

Recommendation, Content, Collaborative Filtering, Learning Automata, Navigation

1. INTRODUCTION

The World Wide Web offers an overwhelming amount of information. Recommendation Systems form a specific technique which attempts to present web pages that are likely of interest to the user. Decision-aid systems like web recommenders are an appropriate means to reduce this abundance of information by filtering out relevant items according to the user's previously stated preferences. Typically, a recommender system compares the user's profile to some reference characteristics. These characteristics may be usage patterns (hits) or from the information present in a website (the content-based approach) or the user's social environment (the collaborative filtering approach)[11].The system compares the collected data to similar data collected from others and calculates a list of recommended web pages for the user. This kind of Recommender system is a useful alternative which will help users discover web pages they might not have found by themselves. Interestingly enough, recommender systems are often implemented using search engines indexing non-traditional data.

Learning automata algorithm for web page recommendations are based on the user’s navigational behavior in a website to discover usage patterns which will generate the recommendations for new users with similar profiles [6]. This method is based purely on the usage of previous user sessions and it doesn’t consider the content of those pages. The connectivity feature of web graph plays an important role in the process of recommendation [8]. The main drawback of Learning automata based recommender algorithm is that the computation of recommendation set is time consuming and limits the algorithms’ performance.

Weighted Association (WA) rule mining is where each web page is allowed to have a weight based on the frequency of visit and duration spent on those web pages. Association rule mining is an important model in data mining. Many mining algorithms discover all web page associations (or rules) in the data that satisfy the user-specified minimum support and confidence constraints. The weights are associated with the web pages to solve the question of different importance of the web pages [5]. The challenge of using weights in the iterative process is being used in generating large frequent itemsets. These item sets are used for recommending the web pages. The problem with WA is the process of matching current users’ session with all of the generated rules needs a lot of time.

Content based (CB) recommendation is one of the methods of recommending web pages. Here web pages are represented as n-grams which compare the frequency of n-gram occurrence present in the current user profile and in the users’ history [1]. Hence this method takes into account the content of the web pages rather than based only on the usage. Systems that recommend web pages to the user based upon a description of the web page and a profile of the user’s interests is content based system [2]. Lack of diversity is one of the limitations in content based approaches.

In Collaborative Filtering (CF) technique, the basic idea is to provide web page recommendations or predictions based on the opinions of other like-minded users. The opinions of users can be obtained explicitly from the users or by using some implicit measures [13]. Collaborative Filtering (CF), the prevalent recommendation approach, has been successfully used to identify users that can be characterized as "similar" according to their logged history of prior transactions. However, the applicability of CF is limited due to the sparsity problem, which refers to a situation that the recommendations are based only on previously rated web pages.

By considering these methods and taking the advantages of both content and collaborative filtering and to improve the efficiency of the above mentioned methods, we **propose a new system** that provides the trustworthiness of websites and the confidence of facts. This system is mainly concerned with analyzing web usage logs, discovering similar web pages from the web logs and making recommendation based on the extracted n-grams. The pages with highest similarity are fed through the truth finder process which finds the trustworthiness of those web pages and the confidence of the facts present in them. While Collaborative filtering is commercially most successful approach for the generation of recommendation set.

The paper is organized as follows: Section 2.1 provides Learning automata based recommendation algorithm. In Section 2.2 the Weighted Association rule has been presented. Content based recommendation algorithm has been discussed in Section 2.3. While Collaborative Filtering have been discussed in detail in Section 2.4. We present our enhanced proposed approach in Section 2.5. Section 3 discusses the performance evaluation of the proposed algorithms compared to other methods. Section 4 concludes the paper with future work.

2. METHODOLOGIES

2.1 LEARNING AUTOMATA (LA)

Use Learning automata is one of the methodologies for recommending web pages. This algorithm includes a finite number of actions that can be performed in a random environment, when a specific action is taken place the environment provides a random response which is either favorable or unfavorable [6]. The objective in the design of the *Learning automaton* is to determine how the choice of the action at any stage should be guided by past actions and responses.

2.1.1Transition Probability Matrix:

The initial step is to construct a web graph using the websites present in the user logs with vertices and edges. Let G = (V, E) where V represents the web pages and E represents the links between them from the page x to y then $(x \rightarrow y) \in E$ [7]. Then a transition matrix P is computed using equation (1).

$$p_{ij} = \begin{cases} \frac{1}{\deg(x_i)} : if\ (x_i \rightarrow x_j) \in E \\ 0 : otherwise \end{cases} \tag{1}$$

Where $\deg(x_i)$ is the number of out links that exists from page u.

2.1.2 Path Probabilities:

Consider that the path traversed by a user is $(p_1 \rightarrow p_2 \rightarrow p_3 \cdots \rightarrow p_k)$. Then the path probabilities for m-order model are computed for the transactions of the user as follows,

$$\Pr(p_1 \rightarrow p_2 \cdots \rightarrow p_k) = \Pr(p_1) \times \prod_{i=2}^{k} \Pr(p_i / p_{i-z} \cdots p_{i-1}) \tag{2}$$

Where $\Pr(\bullet \rightarrow \bullet)$ represents the transition probability value and $\Pr(\bullet)$ represents the page rank $\frac{q(i)}{\sum j \in V\ q(j)}$ where $q(i)$ implies the number of users who visits the page i, V is set of Learning automata [8]. Finally the pages with high probability values are presented to the current user.

2.2 WEIGHTED ASSOCIATION RULE

In this approach each web page p has assigned a weight measure for approximating the degree of interest of a web page to the user [5]. General assumption is that high frequency web pages are of highly interested to the user.

$$F(p) = \frac{NumberOfVisits(p)}{\sum p \in Visited(P)\,(NumberOfVisits(p))} \tag{3}$$

Where F is the frequency, i.e. number of visits of the web page.

$$D(p) = \frac{TD(p)/Len(p)}{\max_{p \in VisitedPag\ e} (TD(p)/Len(p))} \tag{4}$$

$$Weight(p) = Frequency(p) * Duration(p) \tag{5}$$

Where D is the duration, TD is the total duration and len is nothing but the length. Here duration is the time the user spends on the particular page, length of the page is used to normalize the duration i.e. total bytes of the page. After finding the weight for each page according to equation (5), the following are computed in order to recommend the web pages [10].

The weight of each itemset X present in a transaction t is calculated using equation(6) and weights associated with pages present in a transaction w(t_k)can be computed using equation(7).

$$w(X,t) = \begin{cases} \min(w(p_1, p_2, \ldots.., p_k))X \subseteq t \\ 0X \not\subset t \end{cases} \tag{6}$$

Where k is the number of items in the itemset.

$$w(t_k) = \frac{\sum_{i=1}^{|t_k|} w(p_i)}{|t_k|} \tag{7}$$

Where w (p_i) is weight of web page i and $p_1, p_2, \ldots.., p_k$ are set of web pages in the transaction, t_k is the set of transactions in the entire user session. The weighted support count wsp(X) of an itemset X across all transaction can be computed using equation (8).

$$wsp(X) = \frac{\sum_{t_i \in T} w(t_i) * w(X, t_i)}{\overline{w} * \sum_{k=1}^{|t|} w(t_k)} \tag{8}$$

Where $\overline{w}$ is the average weight of all itemsets across all transactions and T is the set of all transactions. Apply apriori algorithm for finding the frequent itemsets. For those itemsets find the weighted confidence using following equation.

$$wconf(X \Rightarrow Y) = \frac{wsp(X \cup Y)}{wsp(X)} \tag{9}$$

Where X and Y are the item sets. Recommendation of web page needs recommendation score. In order to find the recommendation score, we need to compute the following.

$$Dissimilarity(S, r_L) = \sum i{:}r_{Li}{>}0 \left(\frac{2 * (w(s_i) - w(r_{Li}))}{w(s_i) + w(r_{Li})} \right)^2 \tag{10}$$

$$MatchScore\,(S, r_L) = 1 - \frac{1}{4}\sqrt{\frac{Dissimilar\,ity(S, r_{Li})}{\sum_{i:r_{Li}>0} 1}} \tag{11}$$

$$w_i = \begin{cases} weight\,(p_i, r_{Li}), if\ :\ p_i \in r_L \\ 0 : otherwise \end{cases} \tag{12}$$

Where w (r_{Li}) is left hand side of weighted association rule r_L = (w_1, w_2….w_m). where w_i is given in equation (12).The current user session can be given as a vector $S = \{s_1, s_2, \ldots s_m\}$ where w(s_i) is nothing but significance weight, if a user has visited the page p_i in this session, and s_i=0 otherwise. Given the weighted association rule and the active session S, the recommendation score can be computed as follows,

$$\text{Re}\,c(S, X \Rightarrow p) = MatchScore\,(S, X) * wconf\,(X \Rightarrow p) \tag{13}$$

As a result, the web pages with high rec value are presented to the active user.

2.3 CONTENT BASED FILTERING

In order to provide recommendations, we need to generate n-grams for the current user and the user history. An n-gram is a subsequence of n items from a given sequence [3]. It is a type of probabilistic model for predicting the next item in such a sequence [2]. If two strings of real text have a similar vector representation then they are likely to be similar.

After generating n-grams cosine similarity which is a measure of similarity between the user history and the active user profile by finding the cosine of the angle between them [9].

$$Similarity\ = \cos(\theta) = \frac{G.H}{\|G\|\|H\|} \tag{14}$$

Where G and H are usually term weights where

w_i = tf_i * log(n/df_i)

tf_i = number of occurrences of the tf_i (n-gram) in that web page

n = total number of web pages.

df_i= number of web pages in which tf_i (n-gram) appears at least once.

Then the web pages with top n similar values are now recommended to the active user.

2.4 COLLABORATIVE FILTERING BASED RECOMMENDATION

Collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting taste information from similar minded users (collaborating)[12].Through this method of filtering, user groups use and test the web page and provide ratings or vote as a feedback that is relevant to the item and the class in which it falls.

Initially find similar users i.e. nearest neighboring users [4]. Two users are said to be similar minded users if they have at least two commonly rated web pages [1]. For finding similarity between them, Pearson Correlation Coefficient $P_{a,u}$ is being computed using,

$$P_{a,u} = \frac{\sum_{i=1}^{m} (v_{a,i} - \overline{v_a}) \times (v_{u,i} - \overline{v_u})}{\sqrt{\sum_{i=1}^{m} (v_{a,i} - \overline{v_a})^2 \times \sum_{i=1}^{m} (v_{u,i} - \overline{v_u})^2}} \tag{15}$$

Where $v_{a,i}$ is ratings for item i by active user a and $v_{u,i}$ is the ratings for item i by user in history u. $\overline{v_a}$ is the mean ratings by active user a and $\overline{v_u}$ is the mean ratings by user in history u and m is the total number of items.

Then consider n users who have highest similarity with active users as neighbors. Using the neighbors calculate the predictions as shown.

$$p_{a,i} = \overline{v_a} + \frac{\sum_{u=1}^{n} (v_{u,i} - \overline{v_u}) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}} \tag{16}$$

These predictions are used to predict web pages for the active user. The web pages with the highest vote calculated using equation (16) is being presented to the user.

2.5 PROPOSED APPROACH (TDCCREC)

Lots of conflicting information is retrieved by the search engines and the quality of provided information also varies from low quality to high quality. We introduced a new approach which includes methodologies like collaborative filtering, content based filtering and incorporate truth finder to judge the trustworthiness of the web page and confidence of the facts present in the web pages that the system recommends. This approach is shown in Fig.1.

This consists of two main categories.

Category A: If the current users' profile exists, (Collaborative Filtering)

Step 1: Filter out web pages from the user logs using Pearson Correlation Coefficient.

Step 2: Find the nearest neighbors.

Step 3: Present the web pages with highest prediction values to the active user.

Category B: If the current users' profile doesn't exists then (Content based Recommendation with Truth Finder)

Step 1: Generate n-grams (tri-grams) form both the current user and the history

Step 2: Find cosine similarity between current and previous user pages.

Step 3: Apply truth finder algorithm for those pages with high similarity.

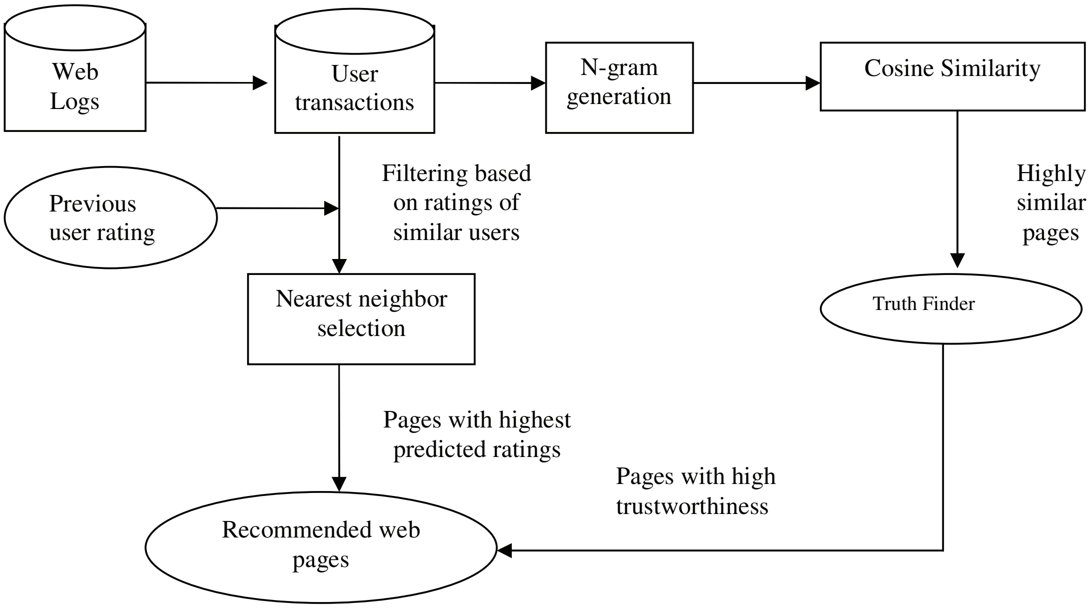
Web Logs
User transactions
N-gram generation
Cosine Similarity
Previous user rating
Filtering based on ratings of similar users
Highly similar pages
Nearest neighbor selection
Truth Finder
Pages with highest predicted ratings
Pages with high trustworthiness
Recommended web pages

Fig.1. System Architecture

2.5.1 Truth Finder:

After finding the cosine similarity, the top n similar pages are fed into this method. This is because, when the information present in those web pages arrive from some source outside the trusted circle, it is much harder to tell whether the certain information is true or not [16]. Thus our proposed (TDCCREC) recommender system not only provides user relevant web pages but also the trustworthy web pages. Our ultimate aim here is to find two parameters namely, confidence of facts and trustworthiness of websites. By finding those parameters, we may able to separate the trustworthy web pages with the corresponding confidence of facts present in them. We have to compute the two parameters only for those websites with greater cosine similarity values. Facts are the properties which describe a particular object or it can be a relationship between two objects (author of a book). For example, an object may be book, movie, music, web page…etc. The facts can be parsed out from the web pages which can be true or false. We can consider some heuristics based on the model which we choose. For example, a fact is likely to be true if it is provided by many trustworthy websites. In this paper we have considered admission dates of Depaul University website as an object. We may also consider the name of the lecturers belonging to a particular department. For example, one webpage may say that the lecturer of CSE department is Harita Bhaskar, while some other may say H.Bhaskar. Sometimes these facts may be complement to each other. Thus we have considered the influences of them too.

Dataset that we have used for our experiment is the preprocessed web logs downloaded from the main DePaul CTI web server (http://www.cs.depaul.edu).The data is based on random sample of users visiting this site for a two week period.

Users may be tempted to distrust things that use a lot of adjectives adverbs and loaded terms [15]. And certainly users should be suspicious. But sometimes people just write that way; it doesn't mean they're lying**.** The main thing is**,** to find the facts i.e. objects. We can check facts. And just ignore the appearances. Trust in any object can be measured by the willingness of visitors to interact with it in some way. When the object is a web page which, means not just looking at the page, but believing the information presented, or acting on it. Trust cannot be totally rational, because it can never be based on enough experience [17]. If a website is trying to convince us to believe one thing but actually talks about another thing, then the website is not trustworthy.

Trustworthiness of websites tw(ws) is the prospected or expected confidence of facts s(f) given by website ws.

$$tw(ws) = \frac{\sum\limits_{f \in F(ws)} s(f)}{\left|F(ws)\right|} \tag{17}$$

Where F(ws) is the set of facts given by ws and tw(ws) is the trustworthiness of the websites ws. Many people may have seen the same source and reported on it themselves. They may have described it differently. We may never know exactly what was said, but if people on different sides of the same issue agree on what was said, then it's more likely to be true.

Confidence of facts s(f) is the probability of fact f to be accurate according to the best of our wisdom.

Case 1: If an object has only one fact then s (f) can be given as

$$s(f) = 1 - \prod_{ws \in WS(f)} (1 - tw(ws)) \tag{18}$$

Case 2: If an object has different facts f1 and f2 then s (f) will be

$$s(f) = \frac{1}{1 + e^{-\gamma\sigma^{*}(f)}} \tag{19}$$

Here the first fact is f and the second fact which is available as some other representation is given as f' .

Where WS (f) is the set of facts provided by the website ws and γ is dampening factor, which is considered as 0.3 here, ρ is weight of objects which controls the influence of the related facts. Its value is between 0 and 1. In order to facilitate the computation, the trustworthiness score of website ws is represented as

$$\tau(ws) = -\ln(1 - tw(ws)) \qquad (20)$$

Where $\tau(ws)$ is the trustworthiness score of the website ws. As this score increases, the trustworthiness of websites too increases. Similarly the confidence score σ (f) of a fact f is shown as in equation (20). Where σ^*(f) is the adjusted confidence score and base_sim is assumed to be a constant value of some threshold level.

$$\sigma(f) = -\ln(1 - s(f)) \qquad (21)$$

$$\sigma^*(f) = \sigma(f) + \rho. \sum_{o(f')=o(f)} \sigma(f') imp(f' \rightarrow f) \qquad (22)$$

$$imp(f_1 \rightarrow f_2) = sim(f_1, f_2) - base\_sim \qquad (23)$$

2.5.2 Algorithm (Truth Finder):

INPUT: The set of web sites WS, the set of facts F, and links between them

OUTPUT: Web site trustworthiness and fact confidence.

```
Calculate matrices A and B (using equation (24) & (25))
For each $ws \in WS$
$tw(ws) \leftarrow t_0$
$\tau(ws) \leftarrow -\ln(1 - tw(ws))$
Repeat
$\overrightarrow{\sigma^*} \leftarrow B\vec{\tau}$
Compute $\vec{s}$ from $\overrightarrow{\sigma^*}$
$\overrightarrow{tw'} \leftarrow \overrightarrow{tw}$
$\overrightarrow{tw} \leftarrow A\vec{s}$
Compute $\vec{\tau}$ from $\overrightarrow{tw}$
Until cosine similarity of $\overrightarrow{tw}$ and $\overrightarrow{tw'}$ is greater than $1 - \delta$
```

Where δ is maximum difference between two iterations and it is set as 0.001 percent.

2.5.3 Calculations and Recommendation Schema for the above Algorithm:

To implement the algorithm, we consider the above mentioned equations such as trustworthiness of websites, trustworthiness score, confidence of facts and confidence scores as vectors i.e. equation (17), (18), (20) and (21).

$$\overrightarrow{tw} = (tw(ws_1) \ldots, tw(ws_M))^T$$

$$\vec{\tau} = (\tau(ws_1) \ldots, \tau(ws_M))^T$$

$$\overrightarrow{\sigma*} = (\sigma*(f_1) \ldots, \sigma*(f_N))^T$$

$$\vec{s} = (s(f_1) \ldots, s(f_N))^T$$

We need to define two matrices A and B. Matrix A for inferring the trustworthiness from the fact confidence and Matrix B for reverse inference purpose.

$$\overrightarrow{tw} = A\vec{s}$$

$$\overrightarrow{\sigma*} = B\vec{\tau}$$

Where A is a x b matrix, a is number of websites and b, number of facts. B is an a x b matrix, which is a transpose of A matrix,

$$A_{ij} = \begin{cases} 1/\left|F(ws_i)\right|, if : f_j \in F(ws_i), \\ 0 : otherwise \end{cases} \qquad (24)$$

$$B_{ji} = \begin{cases} 1, \; if : ws_i \; provides \; f_j \\ \rho.imp(f_k \rightarrow f_j), \; if : ws_i \; provides \; f_k \; and \; o(f_k) = o(f_j) \\ 0 : otherwise \end{cases} \qquad (25)$$

Where o(f) means object that the fact is about. The iteration of the algorithm can be stopped until last two iterations have same values. In our proposed work websites with high trustworthiness value is presented to the active user.

3. EMPIRICAL RESULTS

3.1 DATASETS

The data set contains the preprocessed and filtered sessionized data for the main DePaul CTI Web server (http://www.cs.depaul.edu)[14]. The data is based on a random sample of users visiting this site for a 2 week period during April of 2002. The filtered data files were produced by filtering low support page views, and eliminating sessions of size 1.

3.2 IMPLEMENTATION METHODOLOGY

The original unfiltered data which is downloaded from the DePaul University web server contains a total of 20950 sessions from 5446 users. After preprocessing, the dataset is around 13745 sessions and 683 page views. Also it contains 2734 repeated users. We split our data set as training dataset and test dataset. Here one-third of the dataset is considered for the test set i.e. 4581 user sessions and two-third of the dataset is considered for the training test i.e. 9163 user sessions. The implementation of the entire paper is carried out in the java environment. Here we have presented each user session to the recommender system. We have evaluated our system using some evaluation metrics like similarity, MAD, TW and support. Using which we infer that our proposed system performs efficiently. The expected time complexities for the existing and proposed approaches are worked out in this paper.

3.3 EVALUATION METRICS

The increasing number of web pages retrieved when a query has been posed is an irritating issue for the users when situations arise as which page to be viewed first that have to be indexed in different environments, particularly on the internet. There is a lack of scalability of a single centralized index leading to the use of distributed information retrieval systems to effectively search for and locate the required information with ease.

The performance of the entire system is discussed using metrics like precision, coverage, similarity, and mean absolute deviation, support and transaction weight of the retrieved web pages.

Precision is the fraction of the web pages recommended that are relevant to the user's information need.

$$\text{Precision} = \frac{\left|\{\text{Relevant webpages}\} \cap \{\text{Recommended webpages}\}\right|}{\{\text{Recommended webpages}\}} \tag{26}$$

Coverage is the fraction of the web pages that are relevant to the query that are successfully retrieved.

$$\text{Coverage} = \frac{\left|\{\text{Relevant webpages}\} \cap \{\text{Recommended webpages}\}\right|}{\{\text{Relevant webpages}\}} \tag{27}$$

Similarity is one of metrics used for evaluating the performance of the proposed system. Many measures of similarities are available. We have considered the cosine similarity between the web pages, which can be processed using Eq. (14).

The MAD (Mean Absolute Deviation represents the measurements of the average of the absolute deviations of data points from their mean. Literally, it is the deviation of recommendations from the true user specified values. This can be computed by

$$MAD = \frac{\sum_{i=1}^{N}\left|x_i - \bar{x}\right|}{N} \tag{28}$$

Where x_i is the observed values, $\bar{x}$ is the average and N is the number of values.

Support is the percentage of session in which the page view occurs.

Transaction weight (TW) is a weighting measure calculated from web logs to extract the interest of web pages for the visitor**.** This can be worked out using Eq. (7).

In Table1 sim indicates similarity, MAD is mean absolute deviation, TW is transaction weight, Sup is support and TC is the time complexity.

The time complexity is the amount of time, the system takes to complete the process of execution. Learning automata which is one of the existing approaches takes time O(n^2), where n is the number of nodes present in the web graph. The time complexity of weighted association rule mining is O($n*2^{m-M-N}$) , where m is the number of web pages in the history, n is the number of transactions present in the history, N is the sum of invalid frequent itemsets, M is the sum of frequent itemset. While the time complexity of CF is O($kn^2m/2$), where k is the average selection rate of the training size, n is the number of users and m is the number of items(webpages). The expected time complexity of CB approach is about O(nm), where n is the number of users and m is the number of items i.e. webpages.

The proposed model consumes the time complexity of O(iL+ikn). Here L is the link between the websites and facts, n is the number of facts about each object, k is the average number of facts and i is no the number of iteration involved in the algorithm. In short, the time taken to compute matrix A is O(L) and time taken to compute matrix B is O(kL). Here B contains more entries than A, because B is non-zero if a website w_i provides a fact that is related to the fact f_j. The truthfinder algorithm which involves i iterations takes the O(ikL) time. The time taken to compute website trust worthiness and fact confidence is O(L) and time taken to adjust the fact confidence is O(kn). Finally the overall complexity is considered to be O(iL+ikn).

Table.1. Performance Evaluation of Web Page Recommendation System

Method	SIM	MAD	TW	SUP (%)	TC
LA	0.50	0.68	0.612	55.62	O(n^2)
WA	0.51	0.63	0.637	59.21	O($n*2^{m-M-N}$)
CF	0.81	0.48	0.835	70.45	O($kn^2m/2$)
CB	0.83	0.43	0.856	72.90	O(mn)
CB &LA	0.78	0.52	0.799	69.34	O(n^2)
LA &CF	0.75	0.55	0.786	67.11	O($kn^2m/2$)
CB &CF	0.84	0.40	0.866	75.23	O($kn^2m/2$)
Proposed	0.85	0.30	0.871	78.05	O(iL+ikn)

We have applied these algorithms on standard dataset and experiments shows that our proposed recommender system performs better than the other algorithms and at the same time, proposed system is less complex with respect to memory usage and computational cost too.

Table1 proves that the supplemented evaluation metrics are high for our proposed system. Here metrics like mean absolute deviation, similarity and transaction weight of the retrieved web pages and support are employed.

The mean absolute deviation here shows that the proposed system works well. Because, lower MAD (Mean Absolute Deviation) values indicates that the recommendation system predicts good. The MAD values are more, i.e. around 60% and above for LA and WA systems. This is because both methods gave importance only for the usage patterns. While this is somewhat low for content based and CF systems, as their recommendation is based on content of the web pages and ratings.

As far as the metric similarity is concerned, LA provides only 50% similarity due to the reason that it analyses the previous user logs from the extracted knowledge. While comparing with LA, WA has somewhat better results, because it focuses on duration, frequency and interest of web pages additionally. We have combined the content based and the CF based methods, which has a preferable similarity of about 84%. It can be inferred from Table 1 that our proposed system can perform recommendations significantly, as it gains higher similarity than the other conventional methods.

Also TDCCREC gains higher transaction weight whose parameters are nothing but the frequency and duration. It seems natural to assume that web pages with higher frequency are of strong interest to the user. Support which is one of the evaluation metrics indicates that the proposed system has a good quantity of support count than the other methods. As said earlier, the TDCCREC system has lesser deviation (MAD) with increase in similarity, transaction weight and support.

When content considered as the important parameter in our proposed framework it outperforms the other methods. But we may think that content and usage combined together should have the highest value. But in spite of combining the content and usage methods we have incorporated an efficient algorithm called the truth finder, which gives a hand and increases the efficiency of our proposed work.

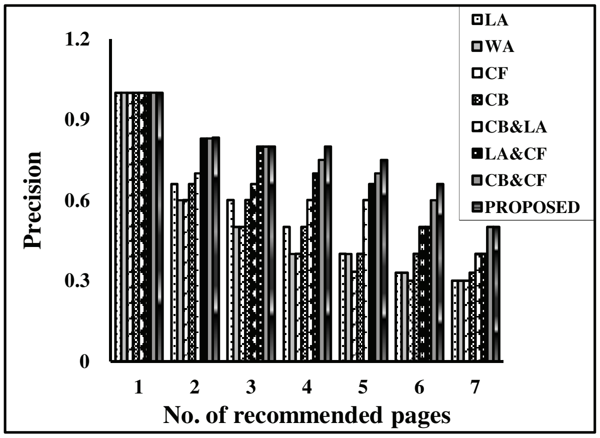

Fig.2. Comparison of proposed algorithm precision with other existing methods

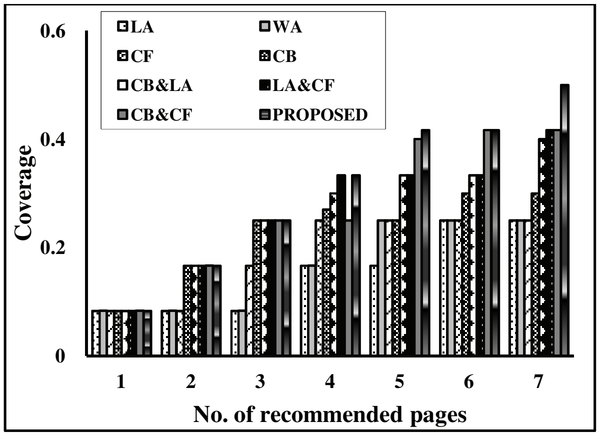

Fig.3. Comparison of proposed algorithm coverage with other existing methods

Hence we have conducted a detailed comparative evaluation of how different combined methods and different recommendation techniques affect the prediction accuracy of the proposed recommender. While giving a glance at the Fig.2 and Fig.3 we come to know that the precision decreases when we increase the coverage, as expected. It shows that precision is inversely proportional to the coverage. As a result, some websites, e.g., those with high link density, may favor a recommender system with high precision, while some others may favor a system with high coverage. The reason for learning automata to gain lowest precision is that it recommends web page merely based only on the usage. Nevertheless it doesn't take the content of the web page into account. It can be concluded that proposed approach is capable of making web recommendation more accurately and efficiently against the conventional methods.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new framework which integrates content based recommendation, collaborative filtering techniques and truth finder. One of the major obstacles for recommendation systems is lack of trustworthiness of websites. Our proposed framework is designed in such a way to deal with problems associated with them and to enhance the performance of existing approaches for web page recommendations. In case of Collaborative Filtering approach, ratings for the web pages are given with the help of experts according to their importance which increases the system behavior. Also we tried for the combination of existing approaches. The experimental results for our framework show that good quality of recommendations can be generated for the active user and most probably matches the users' tastes also; this outperforms the existing techniques for recommendation purpose. Our work provides reasonable accuracy in predictions in the face of high data sparsity.

Our future plan is to perform our current off-line process for web page recommendation into an on-line process to attain greater level of user prediction. We plan to come up with a scenario which pre-calculates and stores the recommendations for each page. We aim to design a system which can respond to new navigation trends and dynamically adapts recommendations for users with suitable suggestions through hyperlinks. In the truth finder algorithm the objects or facts selection must be automated to enhance the performance. In case of CF [1] there is a need to overcome the sparsity or cold start problem.

REFERENCES

[1] M. Balabanovic and Y. Shoham, 1997, "Fab: Content-based, collaborative recommendation," *Communications of the ACM*, Vol.40, No.3, pp. 66.

[2] R. J. Mooney and L. Roy, 2000, "Content-based book recommending using learning for text categorization," *Proc. of the Fifth ACM Conf. on Digital Libraries*, pp. 195.

[3] W. B. Cavnar, 1994, "Using an n-gram-based document representation with a vector processing retrieval model," *In TREC*, pp. 269.

[4] S. S. Anand and B. Mobasher, 2005, "Intelligent techniques in web personalization". *In Lecture notes in artificial intelligence*, pp. 1.

[5] A. Borodin, G.O. Roberts, J.S. Rosenthal, and P. Tsaparas, 2005, "Link Analysis Ranking: Algorithms, Theory, and

Experiments," *ACM Trans. Internet Technology*, Vol.5, No. 1, pp. 231.

[6] Kleinberg, J.M., 1999. Authoritative Sources in a Hyperlinked Environment. *Journals of ACM*, Vol.46, No.5, pp. 604-632.

[7] M. A. L. Thathachar and R. Harita Bhaskar, 1987, "Learning automata with changing number of actions," *IEEE Transactions on Systems Man and Cybernetics*, Vol.17, No.6, pp. 1095.

[8] B. Mobasher. R. Cooley and J. Srivastava, 2000, "Automatic personalization based on web usage mining," *Communications of the ACM*, Vol.43, No.8, pp. 142.

[9] B. Mobasher, H. Dai, T. Luo, Y. Sun, J. Zhu, 2000, "Integrating web usage and content mining for more effective personalization, "*In Proceedings of the First International Conference on Electronic Commerce and Web Technologies*, Springer-Verlag, pp. 165.

[10] H. Ishikawa, T. Nakajima, T. Mizuhara, S. Yokoyama, J. Nakayama, M. Ohta, K. Katayama, 2002, "An intelligent web recommendation system: A web usage mining approach", *In ISMIS*, pp. 342.

[11] P. Resnick, H. R. Varain, 1997. "Recommender Systems," *Communications of the ACM 40*, pp. 56.

[12] T. Hofmann and J. Puzicha, 1999, "Latent class models for collaborative filtering,". In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pp. 688.

[13] M. Deshpande and G. Karypis, 2004, "Item-based top n recommendation algorithms," *ACM Transactions on InformationSystems. Springer-Verlag*, Vol. 22, No.1.

[14] Data sets from Depaul University Server. Available at http://maya.cs.depaul.edu/classes/ect584/data/cti-da-ta.zip. Accessed on 25th December, 2009.

[15] R. Guha, R. Kumar, P. Raghavan and A. Tomkins, 2004, "Propagation of trust and distrust". *In Proceedings of the Thirteenth International Conference on the World Wide Web*, pp. 403.

[16] T. Mandl, 2006, "Implementation and evaluation of quality based Search engine". *In Proceedings of the Seventeenth ACM Conference on HyperText and HyperMedia,* pp. 1.

[17] M. Blaze, J. Feigenbaum, and J. Lacy, 1996, "Decentralized Trust Management". *In Proceedings of the IEEE symposium on Security and Privacy*, pp. 73.