

A Novel Forager Agent Based Approach to Capture the Future Browsing Pattern of Online Users

Mohanraj Vijayakumar (Corresponding author)
Sona College of Technology, Salem-636 005, Tamilnadu, India
E-mail: bvmohanraj@yahoo.in

Lakshmi pathi R
St. Peters Engineering College, Chennai, Tamilnadu, India
E-mail: drrlakshmi pathi@yahoo.com

Abstract

A big challenge for the web master of any e-commerce website is to match user needs and keep them continuously to browse their site. It is easy for any unsatisfied user to reach out the counterpart site in a single click. Many Web usage mining methods were adopted to work on web server log and capture the future browsing pattern of user. However, the accuracy of the methods can't satisfy the user especially in huge site. This paper presents the novel Forager agent based architecture that uses its collective intelligence for identifying the future browsing pattern of user. Our practical implementation shows that accuracy and coverage measures are very much improved than existing methods.

Keywords: Web usage mining, Web intelligence, Artificial bees colony

1. Introduction

The World Wide Web is continuously growing and become de facto standard to conduct online business. In the current internet world, peoples are more attracted towards participating in the online market. There are many e-commerce sites compete with each other to attract the user. It becomes mandatory for web master to predict the future navigation of user (Baraglia & Silversti, 2007) and recommend those to users. This makes the user to browse the site with lot of satisfaction. In case of unhappiness, it's easy for online user to switch over to another e-market site that provides the same kind of service.

All the e-commerce sites are focusing on how to provide the excellent personalized access to users on their sites (Baraglia & Silversti, 2004). The solution of the problem is web usage mining (WUM). WUM is part of web mining which deals with the extraction of knowledge from server log file (Chanchary, Haque & Md.Khalid, 2008). Source data mainly consists of the logs, that are collected when user access web server and might be represented in standard format. WUM (Frias-Martinez & Karamcheti, 2003) has become very critical for website management, creating adaptive website, business and support services, personalization and network traffic flow analysis.

Typically, the WUM based future browsing pattern capturing process can be divided into FrontEnd and BackEnd with respect to the web server activity (Cooley, Srivastava & Mobasher, 2000). The activities of BackEnd component is focused on building the knowledge base by analyzing server log file which records user web usage data. The activities of the FrontEnd component are classifying the current user navigations to any one of cluster formed in BackEnd Phase and infer the useful pattern to predict the future navigations of user. A particular feature of our paper is that achieving the accuracy of excellence in predicting the future browsing pattern of user using the collective intelligence of Artificial Bee System (Tereshko & Loengarov, 2005). This system is relatively new member of swarm intelligence. It tries to model natural behaviour of real honey bee in food foraging. Honey bee use several mechanisms like waggle dance, round dance and tremble dance to exchange the information about location and profitability of food source. This makes them a good candidate for developing new intelligent search algorithms. Artificial Bee system has three area of study: Foraging Behaviour, Marriage Behaviour and Queen Bee concept. Our paper focuses on the usage of foraging behaviour of bee in the FrontEnd Phase of Forager architecture to achieve excellence in capturing the future browsing pattern of online users.

The paper is organized as follows. In section 2, we review the related automatic recommendation systems and reported as literature survey. In section 3 describes the different phases of Forager agent based architecture and section 4 focuses on the Greatest Common Subsequence detection algorithm which is used by foragers and Intuition Deductive inference engine used by the onlooker in the FrontEnd phase of the architecture. In section 5,

the results of our practical implementation are reported. Finally, section 6 concludes our work.

2. Related Works

Recently, a number of studies have been proposed to capture the future browsing pattern of web users and their navigational behaviour. We have conducted investigation on different WUM system and architecture that can be matched with our proposed system.

A Web personalizer system (Mobasher, 1991) provides dynamic recommendation, as a list of hyperlinks to users. In the Web personalizer system, analysis is based on the usage data combined with structure formed by the hyperlinks of site. Aggregated usage profile is obtained by applying data mining technology [i.e clustering, association rule] in pre-processing phase (Nakagawa & Mobasher, 2003). In this phase web server logs are converted into cluster made up of set of pages with the common usage characteristic. The online phase considers the active user session in order to find match among user activities and discover usage profile. Matching entries are then used to compute a set of recommendations which will be inserted into the last requested page as a list of hypertext links. Webpersonalizer is good example two tier architecture for personalized system. However the accuracy of the Webpersonalizer is affected by association rule mining (Agrawal & Srikant, 1996) used for discovery of frequent item set in web log data. The main problem with the association rule mining method is discovery of contradictory association rules. As a result of inconsistent rules, predicting accuracy of system is degraded. Even the non redundant association rule mining algorithm does not help the system because of the web log data nature where the number of page hits is high.

Analog (Yan, Jacobsen, Garcia-Molina & Dayal, 1996) is one of the first web usage mining systems. It consist of online and offline component. The offline component builds session clusters by analyzing user navigation pattern recorded in the log file. In the online component part, active user session is classified according to the generated model. The classification allows identifying the pages matches with the active session and returning the requested page with list of suggestion. Clustering approach of system affected by several limitation especially scalability and accuracy. There is variety of clustering algorithms available for usage. Each approach could have different type of cluster [Exclusive (K-Means), Overlapping (Fuzzy C Means) and Hierarchical].It's difficult to compare the performance of algorithm on large dataset like web log. In addition, Clustering approach used in all recommender system needs to be back up by excellent classification method. Analog did not have the proper classification approach over the overlapping cluster.

Another WUM system called SUGGEST (Baraglia & Palmerini, 2002) provides useful information to optimize web server performance and make easier the web user navigation .SUGGEST adopts a two level architecture composed by an offline creation of historical knowledge and online component that understands the users behaviour .SUGGEST uses the markov model for calculating the probability of a page the web user visit in future after visiting pages in the same session. This system uses the high order markov model (Deshpande & Karypis, 2004) to improve the accuracy. However, the system can't be used for web site made up of large number of pages due to high space complexity. The limitation of system might be a) the memory required to store web server pages is quadratic in the number of pages. It is a sever limitation in the huge web site. b) SUGGEST does not permit us to manage web site made up of pages dynamically generated.

Our survey reveals that there is a race for finding architecture (Jalal, Mustapha, Mamat & Sulaiman.Md, 2008) and classification algorithm to improve the accuracy of capturing future browsing pattern of online users. But still the accuracy does not meet the satisfaction. In our work, we propose advanced Forager agent based architecture and novel user navigation classification approach in the architecture for improving accuracy and space complexity. Our Forager agent based architecture is the inspiration from artificial bee colony introduced by (Karaboga & Basturk, 2008).

- 1) Each employed bee determines a food source, which is also representative of a site, within the neighbourhood of the food source in its memory and evaluates its profitability.
- 2) Each employed bee shares its food source information with onlookers waiting in the hive and then each onlooker selects a food source site depending on the information taken from employed bees. Each onlooker determines a food source within the selected site by herself and evaluates its profitability using the collective intelligence.
- 3) Employed bees whose sources have been abandoned become scout and start to search a new food source randomly (Fluctuation).

The step (2) of the algorithm is realized in the FrontEnd phase of Forager agent architecture. In the FrontEnd Phase, fleet of forager is originated by onlooker agent on number of clusters formed in the BackEnd Phase. Each forager searches the web pages in the cluster based on reinforcement given from onlooker agent. Each forager

executes the Greatest Common Subsequence detection algorithm on its cluster of web pages and also runs the scoring algorithm. The final score of each forager is received by onlooker agent. In case of more than one profitable cluster, it is the onlooker agent that runs the intuition deductive inference engine to choose the best one among the alternatives. Finally, recommends the selected one as future browsing pattern of the user.

3. Different components of Two Tier Forager Agent based Architecture

According to different functionality, our proposed architecture can be divided into two main phase Back End and Front End. Both these phases are tightly coupled with each other and work closely together. The Figure 1 and 2 depicts the BackEnd and FrontEnd architecture of two tier Intelligent Forager Agent respectively. In the Back End Phase there are two main module, Data pre-processing and user navigation mining. The main modules of the Front End phase are onlooker agent, forager agent and Intuition deductive inference engine.

3.1 BackEnd Phase of Architecture

Two main major modules of the BackEnd Phase are Data pre-processing and user navigation pattern mining. In this phase, we perform Data pre-processing on server log to capture navigation session and after that we apply algorithm to mine user navigation pattern. The Detailed module of BackEnd Phase is shown in Figure 3.

3.1.1 Data Pre-processing

The pre-processing of web logs is usually complex and time demanding. It comprises of four different tasks **a) Data collection:** A flat file was constructed from original weblog file. Each record of the file consists of time, ip address, name, requested resource (URL) and HTTP Status code. **b) Data Cleaning:** In this step, we perform the removal of all the data tracked in web log that are useless for mining purpose such as Navigation session performed by robots and web spider. **c) Session Identification and reconstruction:** it involves 1) Identifying the different users session from usually very poor information available in log files and 2) Reconstructing the user's navigation path within the identified session. **d) Content and Structure Retrieving:** Mostly all WUM uses the visited URL as the poor source of information. They do not convey any information about actual page. We employ the content based information to enrich web log data in the form of maintaining ontology (Liu & Keselij, 2007) for each web page in the Array data structure where each index corresponds to page number of web site. This information will be used by the Intuition deductive engine of Front End for the choosing the best classification.

3.1.2 User Navigation Mining

After the data pre-processing, we perform a navigation pattern mining on the identified session. We perform clustering which aims to group session into clusters based on their sharable properties. These patterns will be further used to facilitate the user profiling process of the system. It includes three main modules:

3.1.2.1 Navigation Pattern Modeling

In this step, web pages accessed are modeled as undirected Graph $G = (V, E)$. The set Vertex (V) identifies the different web pages hosted on the web server model. The edge weights are determined by the following equation

$$WP_{ij} = \frac{C_{ij}}{\text{Max} \{C_i, C_j\}} \quad (1)$$

Where C_{ij} is the number of session containing both pages i and j . C_i and C_j are respectively the number of sessions containing only pages i or page j . Dividing by the maximum between single occurrences of two pages has the effect of reducing the relative importance of links involving index pages. Such pages are those that generally do not contain useful content and are used only a starting point for a browsing session. The edge weights (WP_{ij}) are kept in the adjacency matrix WA where each entry WA_{ij} contains the value computed according to equation (1). To limit the number of edge in such graph, elements of WA_{ij} whose value is less than a threshold are known to be less correlated and thus discarded.

3.1.2.2 Clustering

We apply a graph partitioning algorithm to find groups of strongly correlated pages by partitioning the graph according to its connected components. Clusters are formed by starting from a Vertex a DFS on the graph induced by WA is applied to search for the connected component reachable from this vertex. Once the component has been found, the algorithm checks if there are any nodes not considered in the visit. If it so, it means that a previously connected has been split and therefore, it needs to be identified. To do this, DFS is again performed by starting from one of the nodes not visited. In the worst case, when the entire URL in the same cluster, the cost of the algorithm will be linear in the number of edges of the complete graph G . Before the

clusters are put into navigational pattern profile, the clusters are ranked based on values store in the WM matrix. It will be used for classification performed by foragers based on the Greatest Common Subsequence Detection algorithm and also used for knowledge eliciting.

3.2 Front End Phase of Architecture

In the FrontEnd Phase of our system, URL request of the user is processed by the Onlooker Agent (OA) and Forager Agents (FA). In the case of multiple profitable outputs, the best option is chosen by the Intuition Deductive Inference Engine (IDIE) which is run by Onlooker Agent. Finally, captured future browsing pattern is suggested to the user who initiated the URL request to web server.

3.2.1 Work of Onlooker and Foragers Agent in FrontEnd Phase

The critical component of our system is Onlooker agent and Forager agent. Main inputs for these agents are

- 1) Navigation pattern profile: It consists of clusters formed in the BackEnd Phase of our system.
- 2) Live session window: A Sequence $LSW = \{lwp_1, lwp_2, \dots, lwp_m\}$ is the current size of live session window where m is the size of the current active session window.

On receiving URL's in the form of Live Session Window, Onlooker initiates the Foragers that correspond to number of clusters in the navigation profile. Each **Forager agent (FA)** acts on the Navigation profiles by executing the novel Greatest Common Subsequence Detection to discover the subsequence which may be considered as future browsing pattern of web user. Each FA submits the profitable score along the discovered subsequence to the onlooker.

When the **Onlooker Agent (OA)** receives the profitable score along the sub sequences, it starts to decide the best profitable source of navigation profile. In the case of close race between the sub sequences, **Intuition deductive inference engine (IDIE)** plays a crucial role in selecting the right cluster. The main objective of IDIE is to test the each cluster of discovered navigation pattern against already built knowledge base and choose the cluster which attains the maximum number of matches with the knowledge base. Finally, Onlooker suggests the output of IDIE as the future browsing pattern of web user. The cooperation between Onlooker Agent (OA) and Forager Agents (FA) is depicted in the Figure 4. The OA listens to FA. It is just similar to honey bee dancing area where Onlooker listens to dances of different foragers about the profitable food source.

3.2.2 Algorithm for capturing future browsing pattern of user

The following **Algorithm 1** depicts the working behaviour of onlooker and forager agents to capture the forthcoming browsing pattern of user in the FrontEnd phase of the Architecture.

Algorithm 1 capturing future browsing pattern of user using FA and OA

- 1) Live Session Window (LSW) is given as input to the Onlooker Agent. LSW is the set of web page visited by user in the live session. LSW is represented as $\{LWP_1, LWP_2, \dots, LWP_n\}$ where 'n' is the size of session window.
- 2) Onlooker Agent (OA) initiates the number of Forager Agent (FA_i) that corresponds to each cluster in Navigation Profile (NP_n) where 'n' is number of cluster. FA_i - Forager Agent works on ith cluster in NP and $i \in n$. Initially onlooker agent assigns the arbitrary profitable score of value 100 to each **Forager Agent (FA_i)**. The score is denoted as **score(FA_i)**. This score is updated by the respective Forager Agent on the discovery of subsequence in the step 4.
- 4) For each Forage Agent on its assigned cluster of Navigation Profile does the following
 - i. Each FA_i executes the Greatest Common subsequence Detection (described in 3) on its cluster in respect to **live session window (LSW)** which produces the highest degree of GCD as the discovered subsequence which could be the candidate of online user's future browsing pattern.
 - ii. Discovered sequence is denoted as $IBP = \{IBP_1, IBP_2, \dots, IBP_n\}$. Each Forager Agent updates its initial score using Equation (2) and with the help of adjacency matrix WA built in the BackEnd Phase where each entry WA_{ij} contains the value computed according to equation (1).

$$\Delta score(FA_i) = score(FA_i) + \sum_{i=1}^n \sum_{j=1}^m WA_{IBP_i LWP_j} \quad (2)$$

Where $WA_{IBP_iLWP_j}$ = Value in Adjacency matrix between the each page in Future Browsing Pattern (IBP) discovered by Forager and pages in the Live Session Window (LWP).

iii. After each FA_i executed the steps i and ii, Forager Agent sends its updated score and discovered Subsequence to the Onlooker Agent (OA).

5) After receiving the profitable scores from each Forager agents (FA_i), It selects the first 3 High scored Forager Agent's output. The scores are denoted as PS₁, PS₂ and PS₃.

i. Onlooker Agent computes the absolute difference between the PS₁, PS₂ and PS₃ to find the closeness.

$$\text{IF } \left| \text{PS}_1 - \text{PS}_2 \right| \text{ or } \left| \text{PS}_1 - \text{PS}_3 \right| \text{ or } \left| \text{PS}_2 - \text{PS}_3 \right| \leq \beta \text{ Where}$$

β is Uncertain Profitable Threshold value then

There is race between discovered subsequence of PS₁ or PS₂ or PS₃ to become an future browsing pattern of web user. Onlooker Agent (OA) sends the cluster of competing sub sequences to the Intuition Deductive Inference Engine (IDIE). The main objective of IDIE (described later) is to test the each clusters navigation pattern against already built knowledge base and choose the cluster which attains the maximum number of matches with the knowledge base.

ELSE

Onlooker chooses the PS₁'s sequence as best discovered subsequence.

6) Finally, **Onlooker Agent** reports the captured future browsing pattern as PS₁ to user Or best

Sequence selected by the IDIE in the case of competition.

7) Suppose, if the next user activity in live session window different from the suggested captured list

then the system has to restart once again to classify the new user activities.

3.2.3 Greatest Common Subsequence Detection

Every Forager Agent initiated by Onlooker Agent should perform the similarity comparison between set of pages in Live Session Window and web pages in the cluster to discover the subsequence that could be the forthcoming browsing pattern of online user. It's clear that every forager agent has to perform some kind of pattern matching.

In the pattern matching, comparing the similarity between the two sequences a and b are fundamental problem. One of the fundamental problem is to determine the Greatest Common Subsequence (GCS) between a and b . The GCS is a String comparison metric that measures the subsequence of maximum length common to both the sequences. Main objective of Forager Agent (FA) is to find the Greatest Common Subsequence among the sequence of paths in the form of page visits $A = \{a_1, a_2, a_3, a_4, \dots, a_n\}$, $B = \{b_1, b_2, b_3, b_4, \dots, b_n\}$.

Theorem 1 Let $a = \{a_1, a_2, a_3, a_4, \dots, a_n\}$ and $b = \{b_1, b_2, b_3, b_4, \dots, b_m\}$ be the sequences and Let $d = \{d_1, d_2, d_3, \dots, d_n\}$ be any GCD of a and b .

1) If $a_n = b_m$ then $d_1 = a_n = b_m$ and d_{n-1} is a GCD of a_{n-1} and b_{n-1} .

2) If $a_n \neq b_m$ then $d_1 \neq a_n$ implies d is a GCD of a_{n-1} and b .

3) If $a_n \neq b_m$ then $d_1 \neq b_m$ implies d_n is a GCD of a and b_{m-1} .

We have implemented the GCD with added module that outputs the subsequence of indices of the two sequences that match in getting the Greatest Common subsequence. For example,

If $A = \{wp_1, wp_2, wp_3, wp_2, wp_4, wp_2, wp_1\}$ and $B = \{wp_2, wp_4, wp_3, wp_1, wp_2, wp_1\}$. Their GCD is $GCD = \{wp_2, wp_4, wp_2, wp_1\}$.

3.2.4 Combined Effort of Greatest Common subsequence Detection and Intuition Deductive Inference Engine (IDIE)

After performing the clustering algorithm discussed in 3.1.2.2 of BackEnd Phase, We have a set of cluster $wnp = \{wnp_1, wnp_2, wnp_3, \dots, wnp_n\}$ where $wnp_i = \{wp_1, wp_2, \dots, wp_k\}$ is a set of K pages as a users navigational pattern for each $1 \leq i \leq n$. Here, wnp is set of web navigational pattern in the cluster.

We used the navigation profile which has web navigational pattern of different cluster as facts for building the knowledge base of Intuition deductive Inference Engine. In addition, OA an ontology array data structure built in the Content retrieving step of BackEnd phase is used to form the meaningful facts. We used the unique ontological term for each web page in the site. We get the ontology term of each page from Meta tag of each page which conveys the actual content of the page. Also, Structure of Website is used to input the additional facts for knowledge base. In our WUM system, Knowledge base building is considered as critical part.

A Sequence $LSW = \{wp_1, wp_2, \dots, wp_m\}$ is the current size of live session window where m is the size of the current active session window. Before **Onlooker Agent** initiates the Forager Agents to execute GCD on its cluster, we need to order the lsw sequence based on their adjacency weight matrix (WM) constructed in the navigation pattern modeling. Also, we rank all the clusters based on their weight values. Each cluster weight is computed as sum of all its edges weight. After this step, each Forager Agent initiated by OA with the arbitrary profitable score applies the Greatest Common Subsequence Detection on the assigned cluster in respect to **live session window (LSW)** which produces the highest degree of GCD. Each Forager Agent sends its updated score and discovered subsequence to the Onlooker Agent (OA).

After receiving the profitable scores from each Forager agents (FA_i), OA selects the first 3 High scored Forager Agent's output and finds whether absolute difference between them lesser than β (**Uncertain Profitable Threshold value**). In the case of competition between discovered sequences by forager agents, **Intuition deductive inference engine (IDIE)** plays a crucial role in selecting the right cluster among various options.

The main objective of IDIE is to test the each clusters navigation pattern against already built knowledge base and choose the cluster which attains the maximum number of matches with the knowledge base. The rules of knowledge base are written in InterProlog notation. Our inference engine runs on the facts to check how many of web page navigational sequence in each cluster are semantically correct. The rule base is written in such a way that it checks out each cluster against the knowledge base to find the semantic valid matching count. Intuition Deductive Inference Engine is designed to infer the facts by following Top Down inference mechanism. The excerpts of our IDIE knowledge base are shown in Table 1. The fact **Next (wp_7 (cse research), wp_6 (research))** says that wp_7 is the semantically the next page of wp_6 and values in the bracket are ontological terms gathered from OM array. The rule (1) can directly apply on the facts where as rule (2) is recursive nature.

As we discussed earlier, clusters of competing discovered sequences selected by **Onlooker Agent (OA)** is given as input to IDIE. The IDIE infers the count for each cluster that shows how many sub sequences in each cluster are semantically matching with the knowledge base. Finally, IDIE reports a cluster of maximum valid match count. This is used for preparing the intuition captured list and provided to the user as a recommendation. Suppose, if the next user activity in live session window different from the suggested captured list then onlooker agent has to restart the algorithm once again to identify the forthcoming user activities on their site.

4. Illustration of Forager Agent Based System

For our illustration, consider the navigation profile of BackEnd Phase as shown in Table 2. Assume the Live Session Window size as 3 and set of web pages visited by user in the live session as $LSW = \{wp_{37}, wp_{27}, wp_{18}\}$. As stated in our algorithm 1, LSW is given as input to the Onlooker Agent (OA). OA initialize FA_1 (FA_1, FA_2, FA_3, FA_4 and FA_5) that works respectively on Navigation Profile (NP_1, NP_2, NP_3, NP_4 and NP_5) with the initial arbitrary profitable score of 100. Each Forager Agent (FA_i) executes the Greatest Common subsequence (GCD) on its own controlled cluster in respect to Live Session Window (LSW). While Forager Agent discovers the subsequence that could become Future Browsing Pattern (IBP), update its score according to equation (2) i.e. Sum of adjacency matrix values for pages between LSW and IBP is added to initial profitable score. Consider the Table 3 that depicts the score of each Forager Agent (Score (FA_i)) and its discovered sequence (IBP _{i})

Each Forager Agents (FA_i) sends its Final Profitable Score (FA_i) and Discovered Sequence (IBP _{i}) to Onlooker Agent (OA). After receiving the profitable score from all initialized FA_i , OA selects the first three high scored FA_i .

In this example, only FA_3 and FA_5 produced the updated profitable score. The remaining Forager Agents were not updated their initial score. In the next step, Onlooker Agent (OA) finds the absolute difference between $|Score(FA_3) - Score(FA_5)|$ and compares the value with the β (Uncertain Profitable Threshold value).

Here the value of β is assigned to be 50. In this example, the value of $|Score(FA_3) - Score(FA_5)|$ is 41 which lesser than β . This situation is the typical case of competition of who to become the Future Browsing pattern of user between IBP₃ and IBP₅ which are discovered sequence of FA_3 and FA_5 . Now, Onlooker Agent uses the Intuition Deductive Inference Engine (IDIE) to choose the best one among the alternatives IBP₃ and IBP₅.

Onlooker Agent feeds the clusters of competing discovered sequence NP_3 and NP_5 to the IDIE. The main function of IDIE is to check each page of clusters with knowledge base and find whether they are semantically valid next page in respect to pages in Live Session Window. Onlooker Agent chooses the cluster which attains maximum valid match count in respect to LSW. In this example, FA_5 is reported by the IDIE for attaining the more number

of valid matches with knowledge base than FA_3 . From the FA_5 , OA suggest the following list as perfect Future Browsing Pattern of user

$IBP = \{wp_7, wp_{31}, wp_{29}, wp_{26}\}$.

5. Data Analysis Report

Our proposed Forager Agent based system was tested on the web log dataset of Sona College. This site was deployed in the IBM's Web sphere Application Server. All the algorithms of BackEnd and FrontEnd phase were implemented in the JAVA. The knowledge base used in the IDIE of FrontEnd phase was implemented using the InterProlog. InterProlog provides us the ability to call prolog goal through a prolog object and for prolog to invoke any JAVA method through a Java Message Predicate. The BackEnd phase of our system was tested on the weblog entries of 120 students over a period of 8 weeks. There were approximately 52,745 entries in the log file. For our input dataset, BackEnd phase had produced the navigation profile output which consists of 15 clusters. The performance of the proposed system was analyzed based on the two metrics. They are namely Accuracy and Coverage.

5.1 Accuracy based analysis of Forager Agent based System

Accuracy measure is defined as a degree to which captured future browsing pattern as suggested by the system matches with the actual browsing pattern of user. It is given by

$$\text{Accuracy} = \frac{|P(IBP_{np}, LSW) \cap Original_{np}|}{|P(IBP_{np}, LSW)|} \quad (3)$$

LSW - Live Session Window

$P(IBP_{np}, LSW)$ - Navigation Pattern in predicted future browsing pattern of user.

$Original_{np}$ - Original Navigational pattern of user

The Figure 5 depicts the accuracy of our Forager Agent based system as Live Session Window (LSW) size is increased. Our results show that increase of LSW size gives more wisdom to system that improves the accuracy.

The figure 6 depicts the comparison of our Forager Agent based system with the other two recommendation system namely Web personalizer and SUGGEST. Our results show that Our Forager Agent based system outperforms the recommendation system with the excellent behaviour of Forager Agents which uses Greatest Common Subsequence Detection to capture the future browsing pattern of user. In the case of Competition between alternative, Onlooker Agent uses the Intuition Deductive Inference Engine to choose the best one and thereby completely avoids the misclassification in finding the forthcoming browsing pattern compared with other system.

Our Forager Agent based system outperforms the other recommendation system and achieves the accuracy of 92% compare to the Web personalizer (81%) and SUGGEST (83%) when Live size window is 10.

5.2 Coverage based analysis of Forager Agent based System

Coverage measure is defined as the ability of two tier Forager agent based system to produce all page views that are most likely visited by the user. It is given by

$$\text{Coverage} = \frac{|P(IBP_{np}, LSW) \cap Original_{np}|}{|Original_{np}|} \quad (4)$$

The Figure 7 depicts the coverage of our Forager Agent based system as Live Session Window (LSW) size is increased. The practical implementation of our proposed two tier Forager agent based system on sona college dataset prove that there is increase in the accuracy of capturing future browsing pattern of online user and also Coverage measure is excellent.

6. Conclusions

Our proposed two tier Forager agent based system presented in this paper was inspired from onlooker bees in making a decision of profitable food source using a Collective intelligence of Foraging Behaviour in Bee's Hive. In our system, fleet of forager is originated by onlooker agent on number of clusters formed in the BackEnd Phase Each forager executes the Greatest Common Subsequence detection on its cluster of web pages and also runs the scoring algorithm to discover the subsequence that could be the future browsing pattern of user.

Each forager agent sends the final score to onlooker agent. In case of more than one profitable cluster, it is the

onlooker agent that runs the intuition deductive inference engine to choose the best one among the alternatives. Finally, recommends the selected one as future browsing pattern of the user. The practical implementation shows that our approach really improves the accuracy of predicting the forthcoming browsing pattern of user and satisfies the users compared to other systems.

References

Agrawal, R. & Srikant, R. (1996). ‘Mining sequential patterns’, *International Conference on Data Engineering (ICDE)*, Taiwan, 3–11.

Baraglia, R. & Palmerini, P. (2002). “SUGGEST: a Web usage mining system”, *Proceedings of International Conference on Information Technology: Coding and Computing*, 282-287.

Baraglia, R. & Silversti, F. (2007). “Dynamic Personalization of Web Sites Without User Intervention”, *Communication of the ACM*, 63-67.

Baraglia, R. & Silvertri, F. (2004). ”An online recommendation system for large web sites”, *Web Intelligence, IEEE/WIC/ACM, 2004*, 20-24.

Chanchary, F.H., Haque, I., Md.Khalid. (2008). “Web Usage Mining to Evaluate the Transfer of Learning in a Web-based Learning Environment “, *Knowledge Discovery and Data Mining of IEEE*,249-253.

Cooley, R., Srivastava, J., Mobasher, B. (2000). “Automatic Personalization based on Web usage Mining” *Communication of the ACM*, 43(8), 142-151.

Deshpande, M. & Karypis, G. (2004). ‘Selective Markov models for predicting web page accesses’, *Transactions on Internet Technology*, 4(2), 163–184.

Frias-Martinez, E., Karamcheti, V. (2003). ”Reduction of user perceived latency for a dynamic and personalized site using web mining technique”, *WebKDD*, 35-42.

Jalal, Mustapha, N., Mamat, A., Sulaiman.Md. (2008). “OPWUMP -An Architecture for online Predicting in WUM-based Personalization system”, *In 13th International CSI Computer Science*, Springer Verlag.

Karaboga, D. and Basturk, B. (2008). On the Performance of Artificial Bee Colony (ABC) Algorithm, *Applied Soft Computing*, 8(1), 687-697.

Liu,R. & Keselij,V. (2007). ”Combined mining of Web Server logs and Web contents for Classifying user navigation pattern and Predicting users future requests”, *Data & Knowledge Engineering, Elsevier*, 304-330.

Mobasher, B. (1991). Webpersonalizer: A Server Side Recommender System Based on Web Usage Mining. *In Technical Report TR-01-004*.

Nakagawa & Mobasher, B. (2003). ”A hybrid web personalization Model based on site connectivity”, *WebKDD*, 59-70.

Tereshko, V. & Loengarov, A. (2005). Collective decision-making in honey Bee foraging dynamics, *Comput. Inf. Syst. J.* 1352-1372.

Yan, M., Jacobsen, H., Garcia-Molina, & Dayal, U. (1996). "From User Access Patterns to Dynamic Hypertext Linking," *Comp. Networks and ISDN Sys.*, 28(1), 1007-1014.

Table 1. Excerpts of IDIE Knowledge Base

<p>Facts</p> <p>Next (wp₇(cse research),wp₆(research))</p> <p>Next (wp₁₀(it research),wp₆(research))</p> <p>Next (wp₁₇(cse),wp₂₄(course))</p> <p>Next (wp₇(cse research),wp₁₇(cse))</p> <p>Next (wp₂₇(cse staff details),wp₁₇(cse))</p> <p>.....</p> <p>Rules:</p> <p>Subsequence(x,y):- Next(x,y) (1)</p> <p>SuperSubsequence(x,y):- Next(x,z) , Next(z,y) (2).....</p>

Table 2. Navigation Profile of BackEnd Phase

Navigation Profile	Clustered Navigation Pattern
NP ₁	{wp ₂ ,wp ₁₀ ,wp ₁₅ ,wp ₂₀ ,wp ₈ }
NP ₂	{wp ₃ ,wp ₂₇ ,wp ₅₄ ,wp ₁₀₀ ,wp ₁₂₁ }
NP ₃	{wp ₂ ,wp ₁₉ ,wp ₃₇ ,wp ₂₇ ,wp ₃₀ ,wp ₁₈ ,wp ₆₀ }
NP ₄	{wp ₅ ,wp ₁₅ ,wp ₂₃ }
NP ₅	{wp ₇ ,wp ₃₇ ,wp ₃₁ ,wp ₂₇ ,wp ₂₉ ,wp ₂₆ ,wp ₁₈ }

Table 3. Score (FA_i) and its discovered sequence (IBP_i) in respect to Live session window

Forager Agents (FA _i)	Initial Score	Navigation Profile (NP _i)	Discovered Sequence (IBP _i)	Final Profitable Score (FA _i)
FA ₁	100	NP ₁	NO	100
FA ₂	100	NP ₂	NO	100
FA ₃	100	NP ₃	{wp ₁₉ ,wp ₃₀ ,wp ₆₀ }	172
FA ₄	100	NP ₄	NO	100
FA ₅	100	NP ₅	{wp ₇ ,wp ₃₁ ,wp ₂₉ ,wp ₂₆ }	213

NO- No Output.

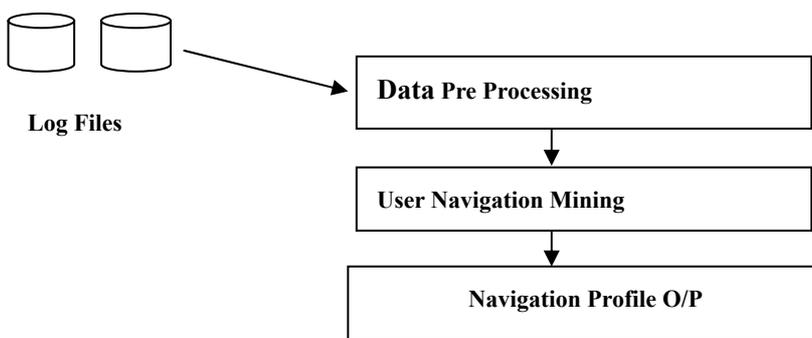


Figure 1. BackEnd Phase of Two Tier Forager Agent Architecture

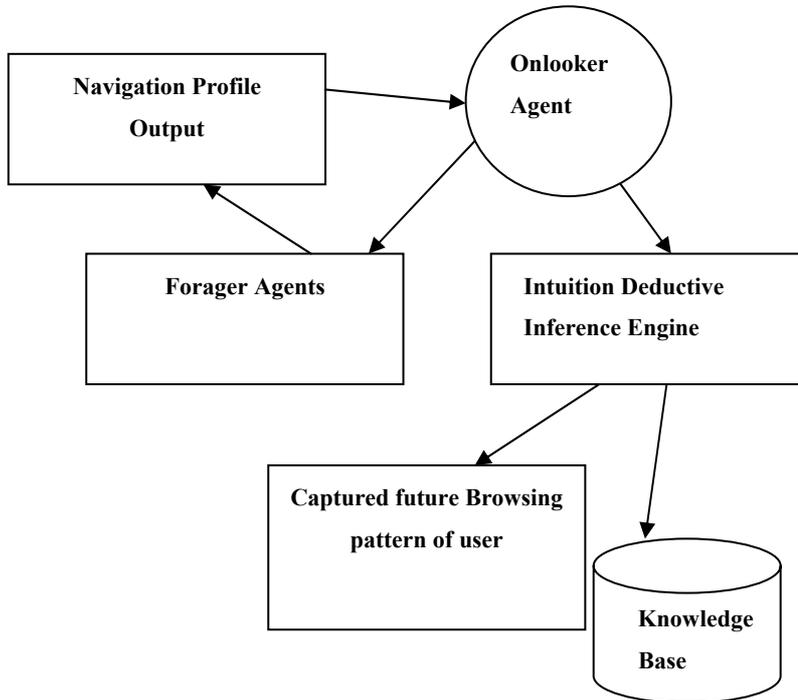


Figure 2. FrontEnd Phase of Two Tier Forager Agent Architecture

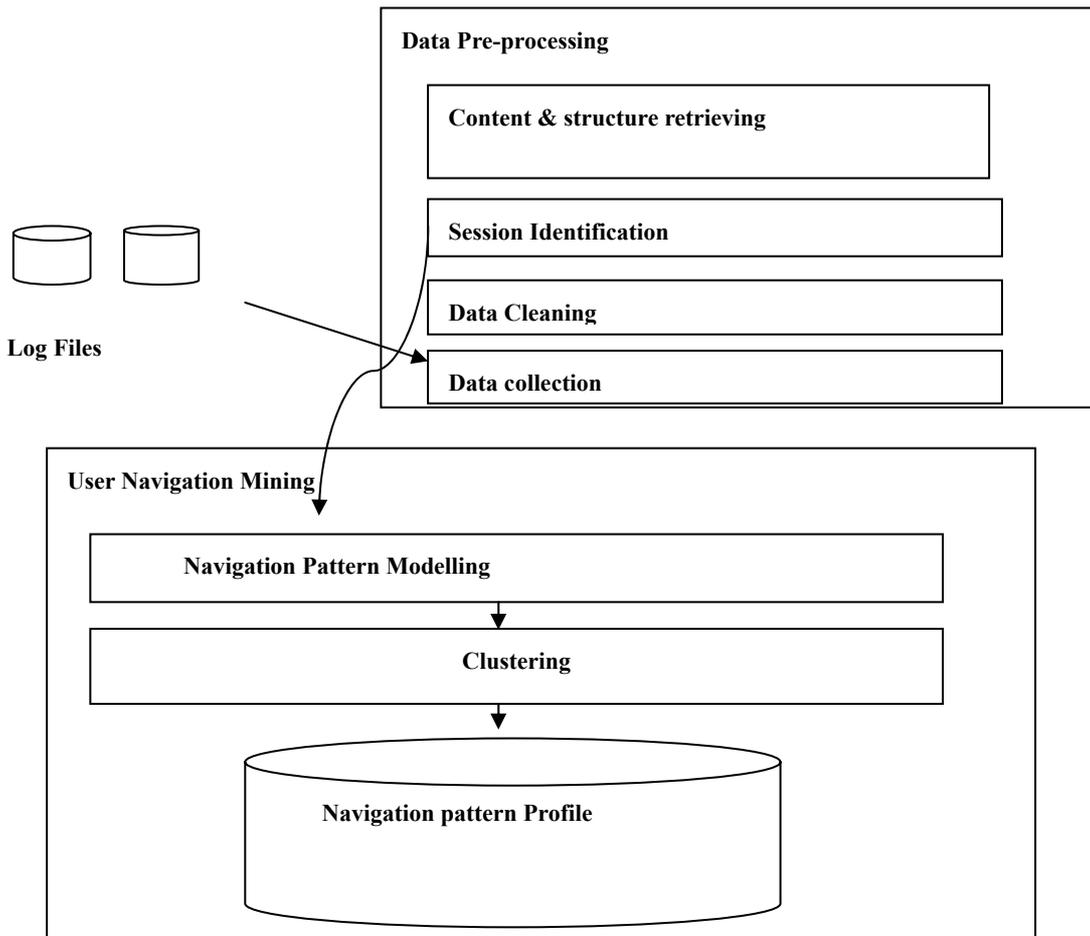


Figure 3. Components of BackEnd

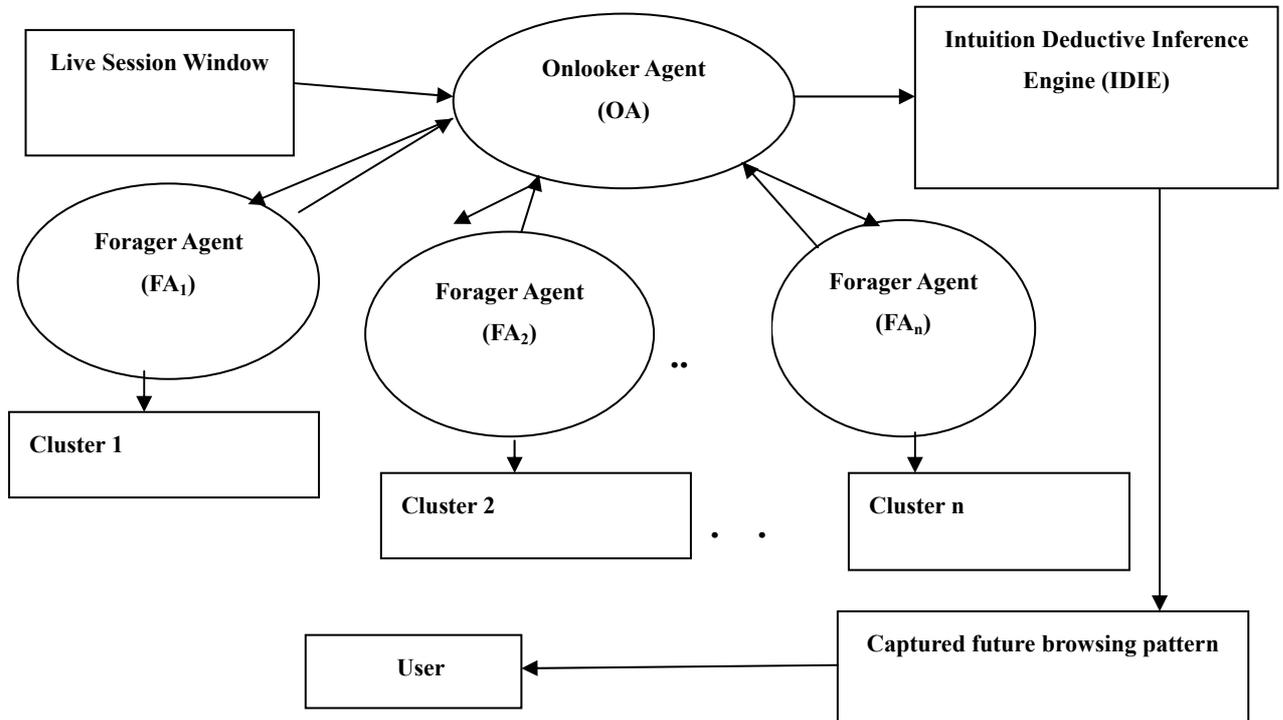


Figure 4. Onlooker Agent and Forager Agents

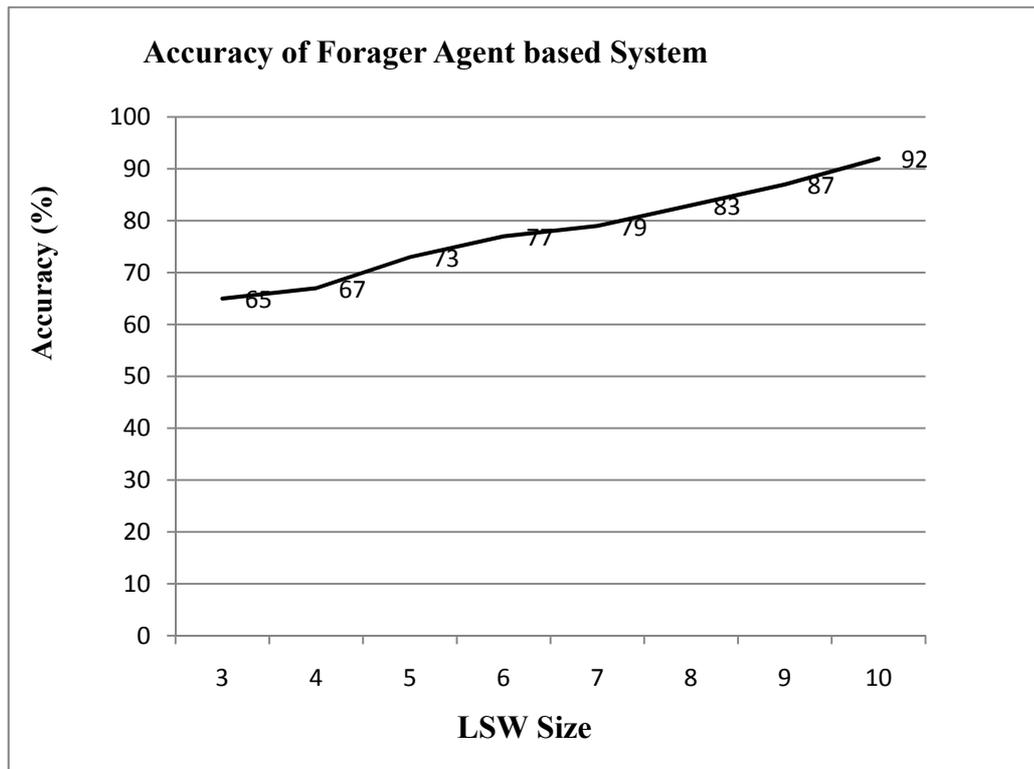


Figure 5. Accuracy of Forager Agent Systems

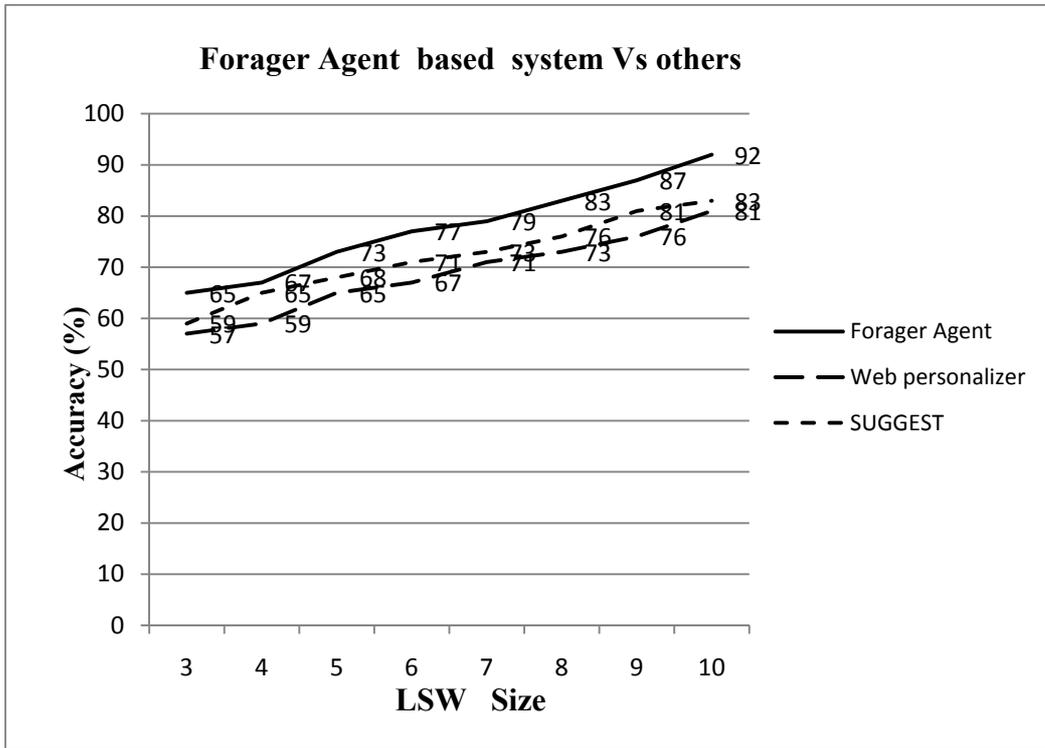


Figure 6. Accuracy of Forager Agent based System Vs Other Systems

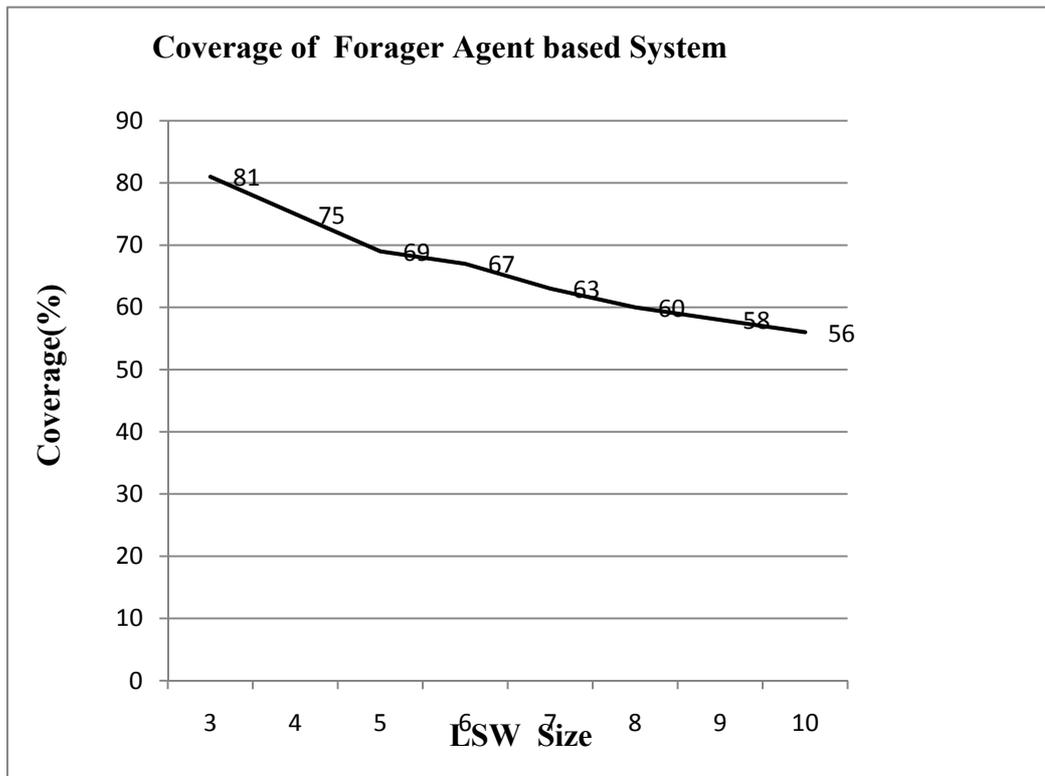


Figure 7. Coverage of Forager Agent based System