Journal of Information Systems Engineering and Management
2025, 10(20s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Applying Optimization Techniques for Enhancing Personalization to Recommend a Web Page

Dr. S. Markkandeyan[1], Dr. I. Kala[2], Dr. M. Rajesh Babu[3], Dr. S. Uma[4], Dr. D. Prasanna[5], M. Senthil Kumar[6]

[1]*Senior Assistant Professor, School of Computing,*
SASTRA Deemed University,
Thanjavur, India
Email: drsmkupt@gmail.com
[2]*Associate Professor, Department of CSE,*
PSG Institute of Technology and Applied Research,
Coimbatore, India
Email: ootykala@gmail.com
[3]*Professor & Associate Dean, School of Computing,*
Rathinam Technical Campus,
Coimbatore, India
Email: drmrajeshbabu@gmail.com
[4]*Professor, PG In charge, Department of CSE,*
Hindusthan College of Engineering and Technology,
Coimbatore, India
Email: druma.cse@hicet.ac.in
[5]*Associate Professor, Department of CSE,*
Mahendra Engineering College,
Mallasamudram, Namakkal (Dt.), India
Email: prasanna@mahendra.org
[6]*PG Student, Department of CSE,*
Hindusthan College of Engineering and Technology,
Coimbatore, India
Email: senthilkumar.kms1997@gmail.com
Corresponding author email id: drsmkupt@gmail.com

ARTICLE INFO

Received: 15 Dec 2024
Revised: 29 Jan 2025
Accepted: 12 Feb 2025

ABSTRACT

Web page recommendation system has been emerging as the most important area in Service computing. Web pages are analyzed and selected for recommendation in order to favor end users while searching for information. Collaborative filtering and content based approaches are two predominant techniques for recommending web pages. Traditional Naive Bayes based probabilistic approach has also shown drastic improvement in achieving personalization during Web page recommendations. However, to improve the accuracy and enhance user satisfaction, we have analyzed optimization techniques such as Ant Colony Optimization and Particle Swarm Optimizations for enhancement of personalization in web search. Here, user profiles comprising of Usage-based and Content-Based attributes are clustered based on similarity in search history. Optimization algorithms are applied to select final web pages from the set of users within the matching cluster. Experiments were carried out with datasets covering 7175 web pages accessed by 287 different users. Result shows that Particle Swarm Optimization outperforms other traditional methods with improved performance.

Keywords: User Profile, Usage-based attributes, Content-Based attributes, Ant Colony Optimization, Particle Swarm Optimization

I. INTRODUCTION

Information retrieval is the process where user-relevant information will be extracted from web servers which are linked with enormous amount of data sources. As the information on the web is increasing day by day, recommendation systems are introduced with the motivation of reducing the search burden of end users [22]. In today's information era, search engines are accessed frequently by web user's for their regular activities. Rather than using traditional searching methods, it is now mandatory to employ artificial intelligence in search engines to

optimally and effectively retrieve the required information [22]. Web page recommendation system is one of the major research areas under web mining that predicts and suggests relevant web pages that are likely to be visited by end users. Hence, the usage of recommendation system reduces delay in search and helps users to achieve the desired purpose in web search.

Personalization in recommender system aims in providing tailored search results in order to increase user satisfaction by creating specific user profile for each web user. User profiles are created by analyzing the user's interest through previous search history and patterns [27]. The web pages that are recommended will be predicted based on these user profiles. The recommender system identifies the similarity among the user profiles by comparing the search and navigation patterns. Top "k" profiles that are ranked based on the similarity with current active user (AU) will be considered for further analysis. The value of "k" depends upon the recommendation engine, which can be fine-tuned to achieve desirable outcome. The web pages that are visited by these top "k" users will be recommended for the current AU.

The main goal of this paper is to reduce the complexity of handling data and to increase the web service recommendation, which simplifies the users' work on giving their preferences (e.g., the user interests and their personal information). The accuracy level of predicting web pages by recommendation systems might be increased by applying optimization techniques and machine learning processes. In this paper we have initially trained the recommendation system using Naive-Bayes based probabilistic algorithm. Here, user profiles comprising of eight Usage-based attributes and two Content-Based attributes are clustered based on the similarity among the profiles. Profile summary will be generated for each cluster which acts as a meta-data for that corresponding group. To improve the accuracy and user satisfaction, we have tried applying optimization algorithms such as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) techniques. We have experimented with various test cases to analyze the effectiveness of these optimization techniques.

The remaining portion of the paper is organized as follows: Section 2 discusses the related work of this paper. Section 3 discusses about the process of creating user profiles using eight Usage-based attributes and two Content-Based attributes. Naïve-Bayes algorithm has been applied for training web recommendation system. Section 4 and 5 narrates the idea of optimizing recommendation using ACO and PSO. Section 6 reports the experimental results. Section 7 gives the concluding remarks and inferences observed. Section 7 gives the concluding remarks.

II. RELATED WORK

Collaborative filtering is one of the most common approaches used for recommendation. Collaborative Filtering systems collect visitor opinions on a set of objects using ratings, explicitly provided by the users or implicitly computed [1]. In explicit ratings, users assign rating to items or web pages, or a positive (or negative) vote to some web pages or documents. The implicit ratings are computed by considering the access to a Web page. A rating matrix is constructed where each row represents a user and each column represents an item or web page keywords. Items could be any type of online information resources in an online community such as web pages, videos, music tracks, photos, academic papers, books etc. Collaborative Filtering (CF) systems predict a particular user's interest in an item using the rating matrix. Alternatively, the item-item matrix, which contains the pair-wise similarities of items, can be used as the rating matrix. Rating matrix is the basis of CF methods. The ratings collected by the system may be of both implicit and explicit forms. Although CF techniques based on implicit rating are available for recommendation, most of the CF approaches are developed for recommending items where users can provide their preferences with explicit ratings to items.

The web log files are collected from the users' browsing history, consisting of IP address, date & time of visiting the web pages, method URL/protocol, status, received byte etc. From the log file all the web page contents are extracted, from which keywords are extracted. Page view and page rank is calculated for each URL. Based on these values, user profile is constructed. The user profile is represented in matrix format. Based on the user profile, user's similarity is found by applying normal recovery similarity measure. Collaborative filtering approach called Normal Recovery Collaborative Filtering (NRCF) is applied on similar users obtained, for web page recommendation [20]. When new user enters a search query same as other similar user query, then the webpages visited by similar users are recommended to the new user.

Content-based filtering is a type of information extraction system, where web pages are extracted based on the semantic similarity between the content in those web pages visited by users in past history [2]. Web content mining applications mostly rely on content-based filtering approaches. Content-based filtering offers predominant support for web page recommendation system. In this technique, the keywords and its frequency of occurrence in those web pages that were previously visited are collected. Then, the semantic similarity between such keywords will be analyzed for further process [2]. For example, consider two users "u1" and "u2" who frequently visit web pages based on their domain of interest. Let u1 always focus on heath related web pages and u2 focus on gadget-related sites. Now, during the real time if any active academic user "ua" search for the query "apple", he will be mostly related to apple devices based sites, rather than apple fruit. So, he will be recommended the sites referred by u2. Similarly, when a dietician "ub" searched for "apple" he will be recommended the sites referred by u1. Recommendation engine classifies "ua" as an academic user and "ub" as a dietician based on the contents (keywords) of the web pages navigated in past history. Along with the keywords, the semantic similarity between them is also analysed for more effective domain grouping. Content-based classification is used for grouping web users under various domains. For such classification, the frequency and keywords in web pages are represented using Term Frequency and Inverse-Document Frequency notations.

Semantic content based approach is another effective recommendation process where semantic similarities between web pages are analyzed [2]. Today many researchers try to combine semantic similarity within the content and collaborative based approaches to improve efficiency. For analyzing the semantic content, user search pattern which are collected from the past history and their personal information acts as implicit and explicit inputs respectively. In many such recommendation systems, explicit inputs that include user's name, user id, area of interest, page likes, feedbacks, etc.., are not considered to be mandatory for predicting web pages that could be further recommended. Latent user preferences are considered where the content alone is not sufficient to find out the interests about the user [6]. Hence overall ratings of web pages are also considered to include unobservable preferences to enhance the recommendation.

The unified and hybrid framework [3] combines both the content and collaborative based approaches along with latent information. The data which present sparsely that are same as the users, interests is hence recommended. In the paper [3] authors propose a generative probabilistic model that incorporates three-way co-occurrence data among users, items and item content which combines both content and collaborative approach. In three-way aspect model users are classified based on the document they access along with the latent variables. Here, the core topic that generates the document retrieval has been considered for computing latent variables. Along with these techniques, k-Nearest Neighbors are used to find the most relevant document which is to be recommended to the new user. Such types of recommendation systems handle the data among the sparse environment [4,5].

Ant Colony Optimization (ACO) is a probabilistic based model [34,35,36] from the family of Swarm Intelligence. ACO employ meta-heuristic methods of optimizations to solve computational problems. It is based on real world phenomena followed by ants in search of its food. The field of bio-inspired computing customizes the phenomena followed by the biological creatures for solving computational problems. In addition to ACO, Particle Swarm Optimization (PSO) is another efficient optimization model from the family of Swarm Intelligence which adapts natural intelligence for computing [39]. The collective behavior of self-organizing particles has been modified suitably for solving computational problems. PSO is also Metaheuristic algorithm, containing a set of algorithms which is used to define heuristic methods. PSO defines a Fitness Function (FF) [39], is also called as Objective Function which expresses the core functionality of research to be optimized. FF could be either in maximization or minimization phenomena.

III. USER PROFILING AND NAÏVE BAYES CLUSTERING

A. Data Preprocessing & Constructing User Profile

The data preprocessing is a stage in which the information about the users are collected. For our research experimentation, we have used AOL log dataset. The log file contains web query log data from 650k users [33]. In order to have privacy preservation, IP addresses of individual users are anonymized. Hence each user is

represented by unique ID. The schema of this log dataset is: {AnonID, Query, Query Time, Item Rank, ClickURL} [33].

Where,

- AnonID – an anonymous user ID number.
- Query – the query issued by the user.
- Query Time – the time at which the query was submitted for search.
- Item Rank – if the user clicked on a search result, the rank of the item on which they clicked is listed.
- Click URL – if the user clicked on a search result, the domain portion of the URL in the clicked result is listed.

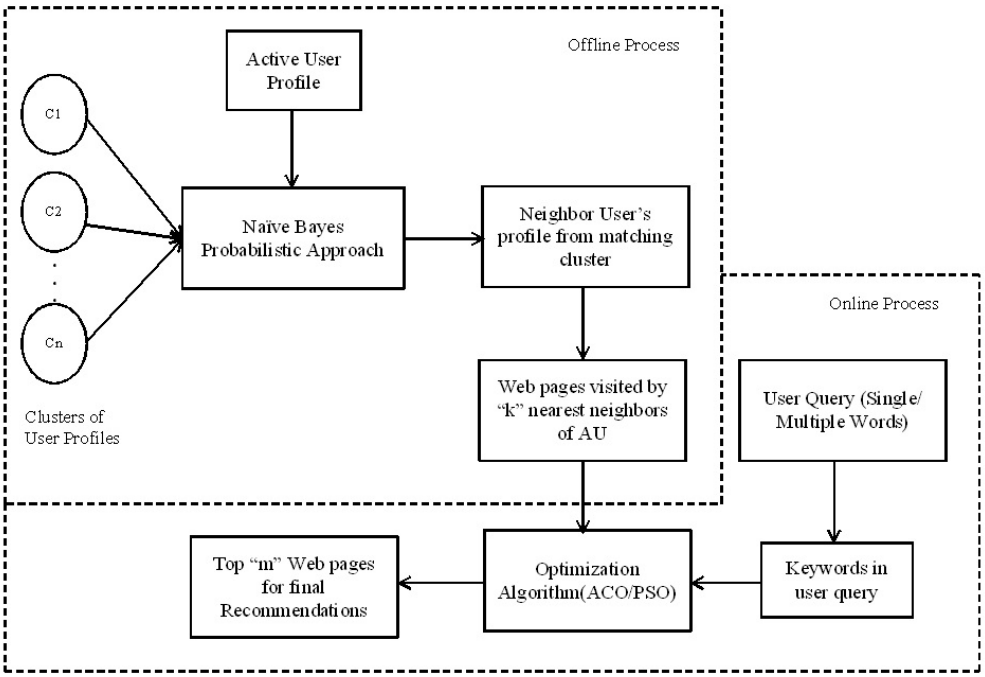


Figure 1 - Recommendation using Naïve-Bayes algorithm

In preprocessing stage, the log file is cleansed by removing unwanted information such as blocked URLs, inappropriate and incomplete entries. Finally, the user profile is constructed by analyzing the search pattern and URLs of each individual user identified using AnonID. A user profile that narrates user interest, searching pattern and web accessing phenomena are created, comprising of the following ten features [32]

- Usage-Based Attributes (8)
    a. Time on Page (TOP)
    b. Time on Site (TOS)
    c. Average Time at this Page (ATP)
    d. Bounce Rate (BR)
    e. Exit Rate (ER)
    f. Conversion Rate (CR)
    g. Number of Visitors (NOV)
    h. Average Page Rank (APR)

- Content-Based Attributes (2)
    a. Top Similar Keywords (SK)
    b. Average Similarity between keywords (ASM)

Weights (β) are assigned for each feature while developing the user profile. The advantage of adding weight is to give more strength to selective features that help in enhancing the accuracy of predicting web pages for recommendation [32]. In the proposed system, the value of β ranges between 1.0 and 2.0. The idea here is to double (β=2) the contribution of most significant features, considerably increase (β=1.75) the strength of significant features, marginally increase (β=1.5) the weight of most relevant features and maintain (β=1.0) the contribution of required features in a user profile to enhance the accuracy of prediction [32]. The following table 1 shows the weight (β) assignment of all features for developing the user profile. Initially, traditional collaborative filtering approach is used to filter "k" number of users (neighbors) from the global set of web users. The value "k" is a level of threshold which can be set by recommendation engine to balance between optimization and increasing search accuracy. The following table 1 states the purpose of each attribute in user profile.

Table 1: List of attributes along with the weightage and significance in creating user profile

Attribute Name	Weight (β)	Description
Time on Page (TOP)	1.00	The total time spent by an active user within a particular page
Time on Site (TOS)	1.00	The time spend by individual user within a particular website
Average Time at this Page (ATP)	1.75	The average time spent by the corresponding user for any page pi (considering various sessions)
Bounce Rate (BR)	1.75	The page pi's access rate between all such sessions is computed as Bounce Rate
Exit Rate (ER)	1.50	The rate at which, the web page (pi) will be at the end of the session is computed as ER
Conversion Rate (CR)	1.75	The conversion rate for each web page is computed as the ratio between total sessions accessed by a user to the total number of sessions that contains the page pi
Number of Visitors (NOV)	1.50	The total number of visitors for each web page has to be computed to analyze the priority of a web page
Average Page Rank (APR)	2.00	Total time spent by the user on particular webpage pi multiplied by number of times that a page is accessed by different users
Top Similar Keywords (SK)	1.75	The top search keywords under each ranked page pi are considered for further recommendation. Top keywords after tokenization and stemming process are considered.
Average Similarity between Keywords (ASM)	2.00	The set of top keywords gathered "k" users are further investigated to find the semantic similarity between each user and current Active User (AU). This similarity is used to find the distance between two users based on their search interest

B. Naïve-Bayes Probabilistic Approach

Naïve Bayes algorithm applies probabilistic based class conditional independence approach for clustering items [7,8]. Feature vectors of known items are used to train the system during clustering. One of the promising aspects of Naïve-Bayesian algorithm is that, it has its independency among each feature [9]. It can effectively consider all the features that are extracted from the users' log file which helps to increase the efficiency of the recommendation to the active use. In the current work, we have extracted the ten features from each user profile in order to train the recommendation system. When any Active User (AU) enters the search query, the profile dataset

of N users who has semantically similar query in past history will be extracted as shown in Table 2. The profile attributes of the active user is also extracted in parallel as shown in Table 3. Naïve Bayes Probabilistic (NBP) algorithm is then applied to cluster these profiles and assign AU into a cluster that contains users whose profiles are similar to AU. Algorithm 1 discusses the steps involved in NBP approach. Finally, NBP identifies nearest neighbors of AU. The overall work flow of the proposed system is shown in figure 1.

Table 2: Profile dataset of N users with similar search query given by AU

Features in User Profile	Assigned Weights (β)	User 1 Profile	User 2 Profile	User 3 Profile	…	User N Profile
UID	NA	841	7895	87	…	785
TOP (In Sec)	1.75	140	126	195	…	183
TOS (in Sec)	1	158	139	187	…	176
ATP (in Sec)	1	58	12	18	…	43
BR (in %)	1.75	0.58	0.49	0.68	…	0.61
ER (in %)	1.5	0.38	0.23	0.53	…	0.41
CR (in %)	1.75	0.0417	0.0256	0.0528	…	0.0394
NOV (in Nos)	1.5	14	6	3	…	9
APR (in Nos)	2	6	1	3	…	4
SK (in Nos)	1.75	254	69	124	…	176
ASM (in Nos)	2	158	69	85	…	248
Cluster ID	NA	3	2	1	…	2

Table 3: Profile with ten attributes of current AU

Features in User Profile	Assigned Weights (β)	Active User (AU)
UID	NA	7999
TOP (In Sec)	1.75	169
TOS (in Sec)	1	153
ATP (in Sec)	1	37
BR (in %)	1.75	0.54
ER (in %)	1.5	0.42
CR (in %)	1.75	0.0423
NOV (in Nos)	1.5	7
APR (in Nos)	2	8
SK (in Nos)	1.75	185
ASM (in Nos)	2	173
Cluster ID	NA	x

Algorithm 1: NBP algorithm for clustering

1. Compute the probability of each cluster's occurrence within the profile dataset as
   P(cluster_x) = Number_of_cluster_x / N
2. Computing Probability Matrix
   For each attribute i = 1 to 10

For all user profiles p = 1 to N

a. Compute the probability of attribute i contributing in classifying the profile p within cluster_x
b. Populate the probability matrix as shown in Table 4 (where three clusters are considered as an example)

3. Identifying the cluster_id for AU
   For each attribute i = 1 to 10 of AU's profile (from Table 3)
   For all clusters c = 1 to M
   a. Compute the probability of mapping AU under each cluster x= 1 to M supported by attributes 1 to 10.

$$P(\text{cluster}=x|AU) = \prod_{i=1}^{10} P(\text{Attribute}_i|\ \text{Cluster}=x) \times P(\text{cluster\_x}) \times \beta$$

   b. Assign AU to the cluster that has maximum probability
4. End Algorithm

IV. ANT COLONY OPTIMIZATION IN RECOMMENDATION

Ant Colony Optimization (ACO) is a probabilistic based model from the family of Swarm Intelligence. ACO employ meta-heuristic methods of optimizations to solve computational problems. It is based on real world phenomena followed by ants in search of its food. Initially ants move in a random manner during the search of their food. While such navigation, they eject a special substance called "pheromone" on their way to food and back to the nest. Other ants following the initial set of ants will never move in random order, instead they follow based on the concentration of the pheromone ejected by ants those reached the food. The concentration of pheromone ejected by the ants that migrates in all other directions opposite to the food will be gradually reduced [34,35,36]. Thus all ants are attracted in a path that optimally reaches the food and back to the nest. The field of bio-inspired computing customizes the phenomena followed by the biological creatures. In this paper, we have implemented the feature of ACO for optimizing the accuracy and delay in prediction of web pages for recommending to the current AU. The functionality of applying ACO in web page recommendation is depicted in Algorithm 2.

Table 4: Probability Matrix derived from profile dataset mapped to three clusters

P(Cluster)	Cluster_1	Cluster_2	Cluster_3
P(TOP | Cluster)	0.36	0.25	0.39
P(TOS | Cluster)	0.40	0.18	0.42
P(ATP | Cluster)	0.45	0.25	0.30
P(BR | Cluster)	0.25	0.39	0.36
P(ER | Cluster)	0.42	0.35	0.23
P(CR | Cluster)	0.36	0.32	0.32
P(NOV | Cluster)	0.33	0.45	0.22
P(APR | Cluster)	0.58	0.23	0.19
P(SK | Cluster)	0.60	0.16	0.24
P(ASM | Cluster)	0.48	0.35	0.17

Algorithm 2: Applying ACO for Optimizing Web Page Recommendation

1. Offline Process:
   a. Identify the web pages ('n') visited by 'k' nearest neighbors obtained through NBP algorithm
   b. Identify the keywords in the web pages ('n') visited by 'k' nearest neighbors and keywords searched by AU using the user profiles
   c. Compute similarity between AU keywords and keywords in web pages
      For each AU keyword j = 1 to m
         For each webpage i= 1 to n
         For each keyword k=1 to si in each webpage i

$$dij = \sum_{j=1}^{m}\left[\sqrt{\sum_{k=1}^{si}\left(Key_j - Key_k\right)^2}\right]$$

   d. Update Pheromone value (P) as shown in Table 5
      For each query keyword j = 1 to m
      For each webpage i= 1 to n

$$Pij = Max(d1j, d2j, d3j, \ldots dij) - dij$$

2. Online Process:
   a. When the user inputs the search query with single/multiple words (m keywords)
   b. Select the web pages that has largest pij value for the corresponding keywords
   c. Recommend the selected web pages.
3. End Algorithm

Table 5: Pheromone Table for Recommendation

Web Pages	Keyword_1	Keyword_2	…	Keyword_m
WebPage_1	P11	P12	…	P1m
WebPage_2	P21	P22	…	P2m
WebPage_3	P31	P32	…	P3m
…	…	…	…	…
WebPage_n	Pn1	Pn2	…	Pnm

V. PARTICLE SWARM OPTIMIZATION IN RECOMMENDATION

Particle Swarm Optimization (PSO) is another efficient optimization model from the family of Swarm Intelligence which adapts natural intelligence for computing [39]. The collective behavior of self-organizing particles has been modified suitably for solving computational problems. PSO is also Metaheuristic algorithm, containing a set of algorithms which is used to define heuristic methods. PSO defines a Fitness Function (FF) which is also called as Objective Function which expresses the core functionality of research to be optimized. FF could be either in maximization or minimization phenomena [37]. The core idea of PSO is to find an optimal solution for FF by searching within a population of potential solutions.

PSO is initialized with a population of random solutions and it gradually searches for optimal one through various generations. PSO also defines a boundary for searching optimal solutions, termed as Search Space (SS). Two key aspects are involved in PSO while finding optimal solutions namely, Social Behaviour and Cognitive Behaviour. The Social Behaviour determines How particle behaves when compared globally (around search space) leading towards

Global Best Solution (GBS). The Cognitive Behaviour determines how particle behaves among themselves (local group of particles) leading towards Local Best Solution (LBS). In each generation of PSO, new Velocity and Position of particles (candidate solutions) will be computed, which makes the generation reaching towards optimal solution.

The third contribution of this paper is to implement PSO for web page recommendation and to check whether this optimizes the performance when compared to traditional recommendations systems and ACO algorithm. The following Algorithm 3 describes the idea of implementing PSO during recommendation.

Algorithm 3: Applying PSO for Optimizing Web Page Recommendation

1. Identify the web pages ('n') visited by 'k' nearest neighbors obtained through NBP algorithm
2. Identify the keywords in the web pages ('n') visited by 'k' nearest neighbors and keywords searched by AU using the user profiles
3. Initialize constants as $\omega = 0.3$, $c_1=0.2$, $c_2=0.2$, $r_1 =1$, $r_2 = 1$, population = n (web pages), velocity for n particles = 0.
4. Assume each webpage as individual particle in the cluster search space. Initialize the position of particles as random values within 100.
5. Compute the Fitness Function (FF) as the following steps
   For each AU keyword j = 1 to m
   For each webpage i= 1 to n
   For each keyword k=1 to si in each webpage i
   $$\text{Min.F(dij)} = \sum_{j=1}^{m} \left[\sum_{k=1}^{si} \left| \boldsymbol{Key}_j - \boldsymbol{Key}_k \right|^{si} \right]^{1/si}$$
6. Evaluate FF as stated in step 4 and check the value of FF that is best among "n" particles.
   a. If best found, stop the algorithm and go to step 9.
   b. Else, proceed with step 7
7. Update Velocity and Position of the particle (web page) as:
   $V_i = \omega V_{i-1} + c_1 r_1 (p_{best} - p_i) + c_2 r_2 (g_{best} - p_i)$
   Where
   P_{best} is local best – the minimum FF among 'n' particles in the current iterations
   g_{best} is global best - the minimum FF among 'n' particles from first to the current iteration
   $P_i = P_{i-1} + V_i$
8. With new position for "n" particles, move to step 5.
9. Recommend those particles (web pages) that are top best solutions among n particles (web pages)
10. End Algorithm.

VI. RESULTS AND DISCUSSION

A. Data Set

For the research experimentation and analysis, AOL log dataset has been used. The log file contains web query log data from ~650k users. In order to have privacy preservation, IP addresses of individual users are represented using anonymous ID. Hence each user is represented by unique ID. The experiments were carried out with datasets covering 7175 web pages accessed by 287 different users. The schema of this log dataset is: {AnonID, Query, Query Time, Item Rank, ClickURL} [33]. Where, Where, AnonID represents an anonymous user ID number to preserve user privacy. Query denotes the query issued by the user. Query Time says the time at which the query was submitted for search. Item Rank denotes that if the user clicked on a search result, the rank of the item on which they clicked is listed. Finally, Click URL represents the domain portion of the URL that the user clicked on a search result. The web access log dataset is divided into seven samples of equal size with 50 records as mentioned in the following table.

Table 6: Various sample datasets used for experimentation

Sample Category	Description
Sample 1	Without any conditions, access log of 50 users were selected randomly
Sample 2	Uniform sampling was performed to select one user after each 50 records.
Sample 3	The query was analyzed and categorized into various domains. 50 users accessed under academic category were selected.
Sample 4	The top 50 users who access web frequently were selected based on the maximum length (no. of URLs) within each session.
Sample 5	The top 50 users who do not access web frequently were selected based on the minimum length (no. of URLs) within each session.
Sample 6	The top 50 users having profile with maximum number of identical search keywords were selected.
Sample 7	The top 50 users having profile with minimum number of identical search keywords were selected.

B. Evaluation Metrics

In order to verify the performance of the proposed algorithm, the following metrics were identified: F1-Measure, Miss-Rate (MR), Fallout Rate (FR) and Matthews Correlation [31]. In order to compute these evaluation metrics, the following table is developed.

Table 7: Contingency table used to compute Precision and Recall

Category	Remarks
True Positive (TP)	The web pages that are recommended were relevant
False Positive (FP)	The web pages that are recommended were irrelevant
True Negative (TN)	The web pages that are not recommended were irrelevant
False Negative (FN)	The web pages that are not recommended are relevant

F1-Measure: The F1-Measure is computed based on two metrics such as Precision or True Positive Accuracy (Confidence) and Recall or True Positive rate [27]. The Precision is defined as the ratio between the recommended web pages that are relevant to the user query to the total number of recommended items. Precision is represented using equation (1).

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (1)$$

Recall is calculated as per equation (2) and is defined as the ratio of web pages recommended that are relevant to the total number of relevant webpages [31] considered for experimentation purposes

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (2)$$

The specifications of TP, FP, TN and FN are stated in Table 8 [31]. These Precision and recall values are used to compute F1-measure as given in equation (3).

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Miss Rate (MR): The Miss Rate is calculated based on the total number of relevant web pages that were not recommended [31]. This is also termed as False Negative Rate as denoted in equation (4).

$$\text{Miss Rate} = \frac{FN}{TP+FN} \quad (4)$$

Fallout Rate (FR): The false positive rate or Fallout Rate is defined as the rate of irrelevant pages that were recommended to the total number of irrelevant pages [31]. This is computed using equation (5)

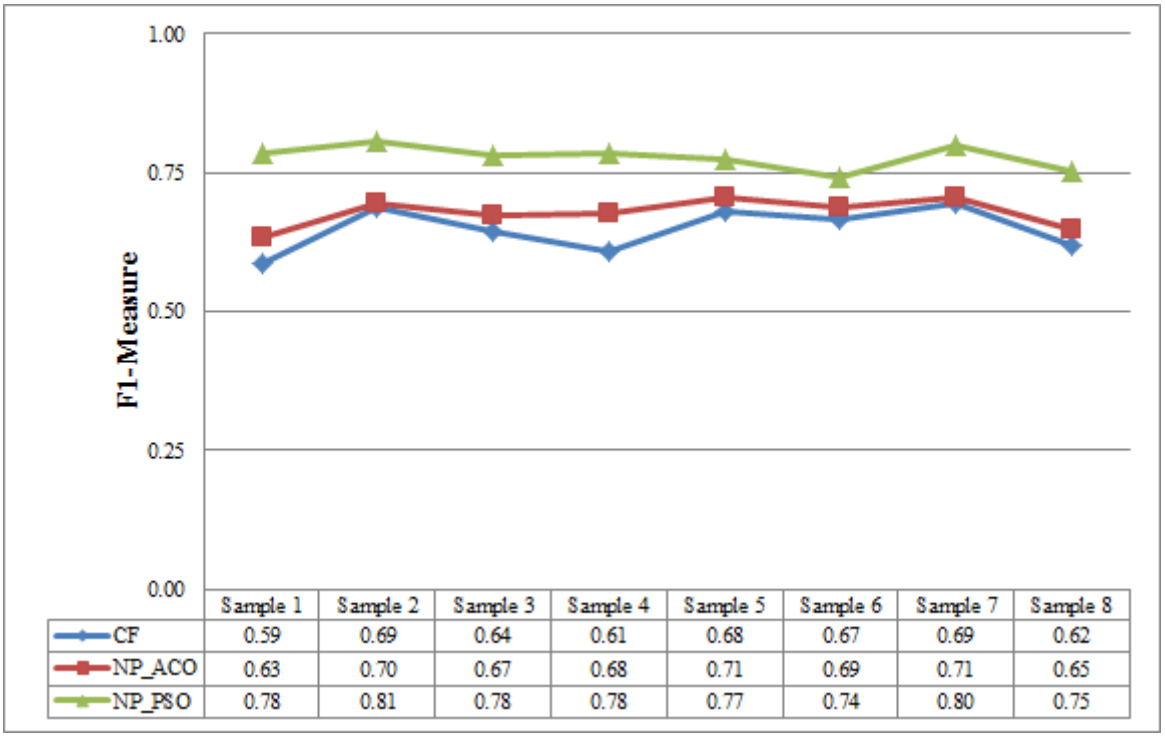
$$\text{Fallout Rate} = \frac{FP}{FP+TN} \quad (5)$$

Matthews Correlation (MC): The Matthews Correlation is used to analyze the effectiveness of the proposed classification and optimization algorithms [31]. This is computed using the equation (6).

$$\text{Matthews Correlation} = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP+FN) \cdot (FP+TN) \cdot (TP+FP) \cdot (FN+TN)}} \quad (6)$$

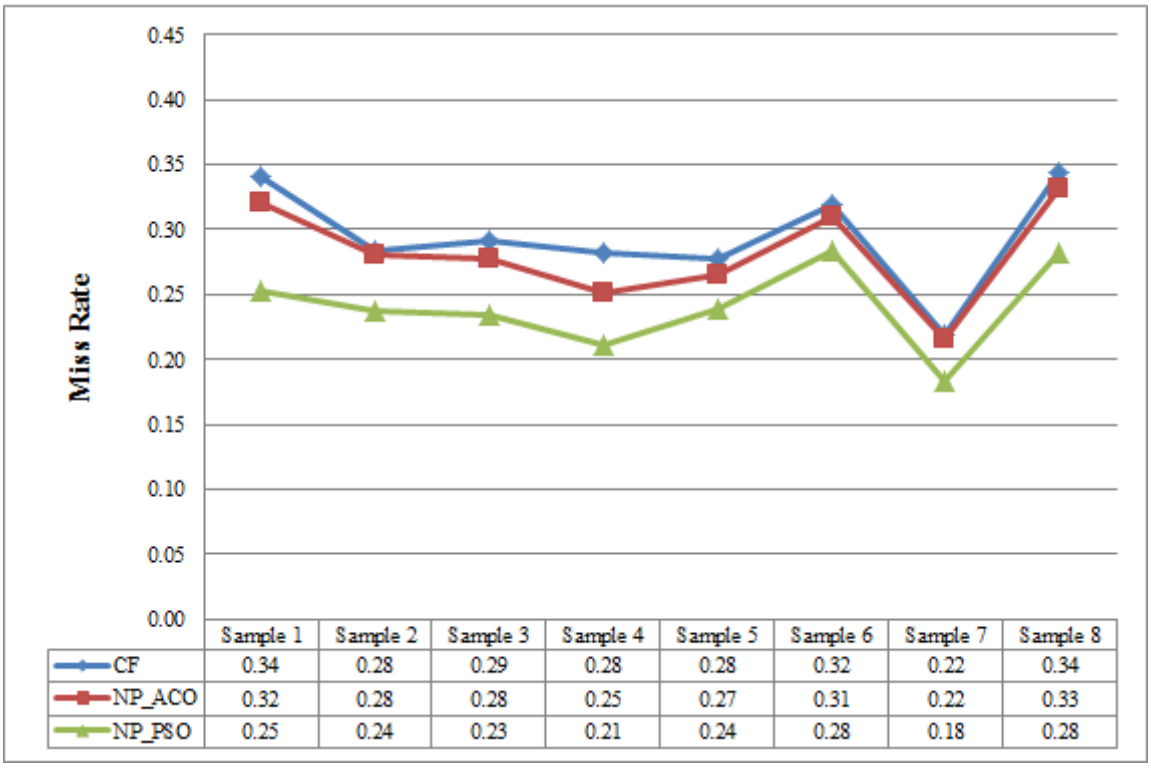
B. Results and Inferences

Experiments were conducted using the seven samples of dataset running under three algorithms Collaborative Filtering (CF), Naïve Bayes Probability with ACO (NP_ACO), Naïve Bayes Probability with PSO (NP_PSO). The graphs that measure F1-Measure, Miss Rate (MR), Fallout Rate (FR) and Matthews Correlation (MC) were shown in figure 2, figure 3, figure 4 and figure 5 respectively. The results clearly depicts that the proposed Naïve Bayes Probabilistic Model with Particle Swarm Optimization has shown improved F1-measure, hence the accuracy is highly maintained. Meanwhile the Miss Rate has been considerably reduced when compared to Naïve Bayes with Ant Colony Optimization technique and traditional collaborative approach in all data samples. The proposed algorithm also shows improvement in classification accuracy, hence while testing under all data samples Matthews Correlation was improved much better for proposed optimization based machine learning classification approach.



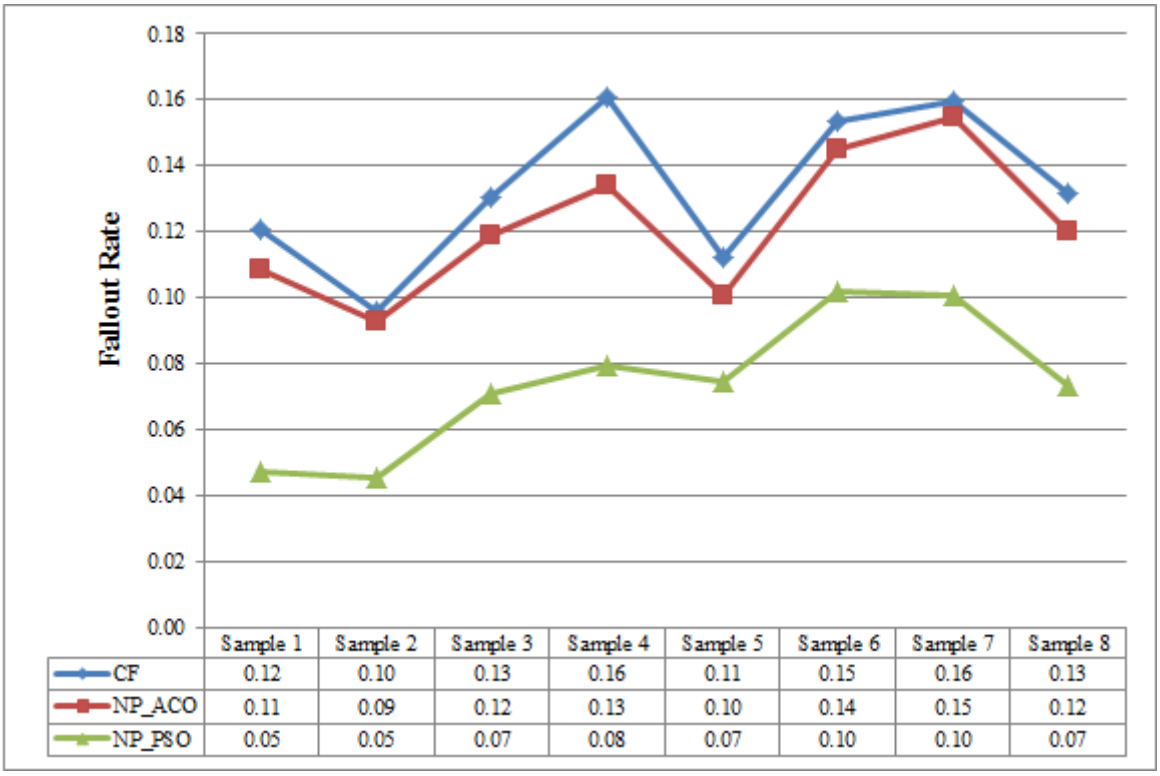
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8
CF	0.59	0.69	0.64	0.61	0.68	0.67	0.69	0.62
NP_ACO	0.63	0.70	0.67	0.68	0.71	0.69	0.71	0.65
NP_PSO	0.78	0.81	0.78	0.78	0.77	0.74	0.80	0.75

Figure 2 – Analyzing F1-measure tested with various sample datasets



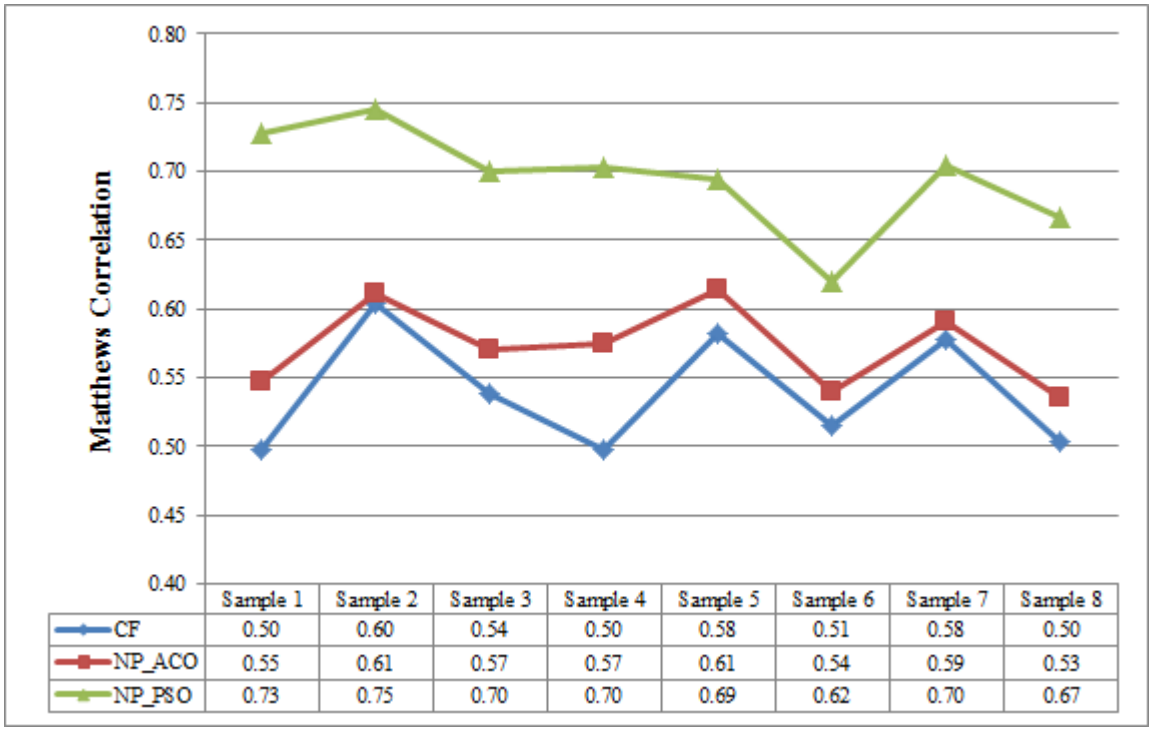
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8
CF	0.34	0.28	0.29	0.28	0.28	0.32	0.22	0.34
NP_ACO	0.32	0.28	0.28	0.25	0.27	0.31	0.22	0.33
NP_PSO	0.25	0.24	0.23	0.21	0.24	0.28	0.18	0.28

Figure 3 – Analyzing Miss Rate tested with various sample datasets



	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8
CF	0.12	0.10	0.13	0.16	0.11	0.15	0.16	0.13
NP_ACO	0.11	0.09	0.12	0.13	0.10	0.14	0.15	0.12
NP_PSO	0.05	0.05	0.07	0.08	0.07	0.10	0.10	0.07

Figure 4 – Analyzing Fallout Rate tested with various sample datasets



	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8
CF	0.50	0.60	0.54	0.50	0.58	0.51	0.58	0.50
NP_ACO	0.55	0.61	0.57	0.57	0.61	0.54	0.59	0.53
NP_PSO	0.73	0.75	0.70	0.70	0.69	0.62	0.70	0.67

Figure 5 – Analyzing Matthews Correlation tested with various sample datasets

VI. Conclusion

In this paper, a novel approach to develop user profiles was proposed where eight usage-based attributes and two content-based attributes were used for efficient characterization of web user profiles. In addition, a new algorithm based on Naive Bayes Probabilistic approach was proposed to cluster the user profiles based on their common interest and web usage pattern. In addition, the effectiveness of applying optimization algorithms for web page

recommendation was also analyzed. We customized the idea of Ant Colony Optimization and Particle Swarm Optimization techniques and analyzed their performances. Experiments were conducted with eight categories of test samples. Results infer that, the Particle Swarm Optimization outperforms when compared to traditional collaborative filtering approach and ACO algorithms with improved F1-Measure. The Miss Rate and Fallout Rates were also found to be decreased, hence enhancing the accuracy. Matthews Correlation value is found to be improved while applying PSO based optimization rather than ACO.

References

[1]. Zibin Zheng, Hao Ma, Michael R. Lyu, Fellow, and Irwin King, (2011), "QoS-Aware Web Service Recommendation by Collaborative Filtering", IEEE Transactions on Service Computing, Vol.4, no.2, pp.140–152.

[2]. Freddy L´ecu´e,(2010), "Combining Collaborative Filtering and Semantic Content-based Approaches to Recommend Web Services", IEEE Fourth International Conference on Semantic Computing, pp. 200-205.

[3]. Alexandrin Popescul Lyle H. Ungar, David M. Pennock, Steve Lawrence, (2001), "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments", Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, pp. 437-444.

[4]. Byron Bezerra and Francisco de A. T. E Carvalho, (2004), "A Symbolic Hybrid Approach to Face the New User Problem in Recommender Systems", Springer Verlag Berlin Heidelberg, pp. 1011–1016.

[5]. Katja Niemann and Martin Wolpers (2015), "Creating Usage Context-Based Object Similarities to Boost Recommender Systems in Technology Enhanced Learning", IEEE Transactions on Learning Technologies, Vol. 8, no. 3, pp. 274-285.

[6]. Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, Hiroshi G.O kuno, (2006), "Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences", University of Victoria.

[7]. Meghna Khatri, (2012), "A Survey of Naïve Bayesian Algorithms for Similarity in Recommendation Systems" , International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, pp. 217-219.

[8]. Kebin Wang and Ying Tan, (2011), "A New Collaborative Filtering Recommendation Approach Based on Naïve Bayesian Method", Springer Verlag Berlin Heidelberg, pp. 218–227.

[9]. Mustansar Ali Ghazanfar and Adam Pru¨gel-Bennett, (2004), "An Improved Switching Hybrid Recommender System Using Naïve Bayes Classifier and Collaborative Filtering", School of electronics and Computer Science, University of Southampton, United Kingdom.

[10]. Mingming Jiang, Dandan Song, Lejian liao, Feida Zhu, (2015), "A Bayesian Recommender Model for User Rating and Review Profiling", Tsinghua Science and Technology, pp 634-643.

[11]. Xi Chen, Xudong Liu, Zicheng Huang, and Hailong Sun, (2010), "A Scalable Hybrid Collaborative Filtering Algorithm for Personalized Web Service Recommendation", IEEE International Conference on Web Services, pp-9-16.

[12]. Jonathan L. Herlocker and Joseph A. Konstan, Loren G. Terveen, and John T. Riedl, (2004), "Evaluating Collaborative Filtering Recommender Systems", ACM Transactions on Information Systems, Vol. no. 22, pp.5-53.

[13]. John Z. Sun, Dhruv Parthasarathy, and Kush R. Varshney, (2014), "Collaborative Kalman Filteringfor Dynamic Matrix Factorization", IEEE Transactions On Signal Processing, Vol.62, pp.3499-3509.

[14]. J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez (2013),"Recommender systems survey" Published in Elsevier,Universidad Politécnica de Madrid, Ctra. De Valencia, Spain, pp.109-132.

[15]. Mustansar Ali Ghazanfar and Adam Prugel-Bennett,(2010). "A Scalable, Accurate Hybrid Recommender System", IEEE Third International Conference on Knowledge Discovery and Data Mining, pp. 94-98

[16]. Lina Yao, Quan Z. Sheng, Member, IEEE, Anne. H.H. Ngu, Jian Yu, and Aviv Segev(2015), "Unified Collaborative and Content-Based Web Service Recommendation", IEEE Transactions On Services Computing, Vol. 8, No. 3 pp.453-466.

[17]. Zheng Lu, Hongyuan Zha, Xiao kang Yang, Weiyao Lin, Zhaohu iZheng (2013), "A New Algorithm for Inferring User Search Goals with Feedback Sessions", IEEE Transactions on Knowledge and Data Engineering, VOL. 25, NO. 3, pp. 502-513.

[18]. Dimitrios Pierrakos, Georgios Paliouras, “Personalizing Web Directories with the Aid of Web Usage Data”, IEEE Transactions on Knowledge and Data Engineering, VOL. 22, NO. 9,pp. 1331-1344.
[19]. D. Ciobanu, C. E. Dinuca (2012), “Predicting the next page that will be visited by a web surfer using Page Rank algorithm”, International Journal of Computers And Communications, Issue 1, Volume 6, pp: 60-67.
[20]. Huifeng Sun, ZibinZheng, Junliang Chen and Michael Lyu, R. (2013), “Personalized Web Service Recommendation via Normal Recovery Collaborative Filtering”, IEEE Transactions on Services Computing, vol. 6, pp.573-579.
[21]. Daxa k. Patel (2014), “A Retrieval Strategy for Case-Based Reasoning using USIMSCAR for Hierarchical Case”, International Journal of Advanced Engineering Research and Technology, Volume 2 Issue 2, pp. 65-69
[22]. Şule Gunduz-Oguducu, M. Tamer Ozsu, (2006), “Incremental click-stream tree model: Learning from new users for web page prediction”, Distributed Parallel Databases, Springer Science, Vol: 19: 5–27.
[23]. Amit Vishwakarma, Kedar Nath Singh, (2014), “A Survey on Web Log Mining Pattern Discovery”, International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 7022-7031.
[24]. Tranos Zuva, Sunday O. Ojo, Seleman M. Ngwira and Keneilwe Zuva (2012), “A Survey of Recommender Systems Techniques, Challenges and Evaluation Metrics”, International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 11, pp.382-386.
[25]. Aditi Shrivastava, Nitin Shukla(2012), “Extracting Knowledge from User Access Logs”, International Journal of Scientific and Research Publications, Volume 2, Issue 4.
[26]. G. Schröder, M. Thiele, and W. Lehner (2011), "Setting goals and choosing metrics for recommender system evaluations," in Proceedings of the Second Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI 2).
[27]. Murat Goksedef, ŞuleGunduz-Oguducu (2010), “Combination of Web page recommender systems”, Elsevier Journal on Expert Systems with Applications, 2911–2922.
[28]. Gediminas Adomavicius, Alexander Tuzhilin, (2005), “Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art andPossible Extensions”, IEEE Transactions on Knowledge and Data Engineering, VOL. 17, NO. 6, pp:734-749.
[29]. ThiThanh Sang Nguyen, Hai Yan Lu, and Jie Lu (2014), “Web-Page Recommendation Based on Web Usage and Domain Knowledge”, IEEE Transactions On Knowledge And Data Engineering, VOL. 26, NO. 10,pp: 2574-2587.
[30]. Pearl Pu, Li Chen,Rong Hu (2012), “Evaluating recommender systems from the user’s perspective: survey of the state of the art”, Springer Science vol. 22; pp:317–355.
[31]. Schröder, G., Thiele, M., Lehner, W. (2011), “Setting Goals and Choosing Metrics for Recommender System Evaluations”, Second Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces.
[32]. Abirami, S., Bhavithra, J., Saradha, A. (2017), “A Web Page Recommendation using Naive Bayes Algorithm in Hybrid Approach”, IEEE International Conference on Science, Technology, Engineering and Management, March 2017.
[33]. Michael, G. Noll. (2006)., “AOL Research Publishes 650,000 User Queries”, URL:http://www.michael-noll.com/blog/2006/08/07/aol-research-publishes-500k-user-queries/
[34]. Weihui Dai, Shouji Liu and Shuyi Liang. (2009), “An Improved Ant Colony Optimization Cluster Algorithm Based on Swarm Intelligence”, Journal of Software, VOL. 4, NO. 4,pp.299-306.
[35]. Pablo Loyola, PabloE.Roma, JuanD.Vela´squez. (2012), “Predicting web user behavior using learning-based ant colony optimization”, Elsevier - Engineering Applications of Artificial Intelligence 25, pp.889-897.
[36]. Xiao-Feng Xie, Wen-Jun Zhang, Zhi-Lian Yang. (2002), “A Dissipative Particle Swarm Optimization”, IEEE Congress and Evolutionary Computation, pp~1456-1461.
[37]. Renato A. Krohling. (2004), “Gaussian Swarm: A Novel Particle Swarm Optimization Algorithm”, IEEE Conference on Cybernetics and Intelligent Systems, pp~372-376.
[38]. Deepa S.N., G. Sugumaran. (2011), “Model order formulation of a multivariable discrete system using a modified particle swarm optimization approach”, Elsevier: Swarm and Evolutionary Computation, pp~404-212.