Abstract—Recommender systems are the software agents which are widely used to handle the problem of information overload. Most recommender systems employ collaborative-filtering or content based methods to suggest new items of interest for users. Both these methods have complementary advantages and disadvantage but independently they fail to provide good recommendations in many circumstances. In this paper we discuss standard architectural framework Semantic Enhanced Personalizer (SEP) which combines three recommendation techniques i.e., original, semantic and category based. This framework overcomes problem of existing recommender systems such as cold-start and sparsity. We also evaluate performance of Apriori and Performance Based Transposition Algorithm for frequent Itemset generation from real life dataset. From the experiments, it is shown that PBTA is much faster than Apriori algorithm and it drastically reduce CPU and I/O overhead. Finally, we implement category based recommendation module of SEP Architecture using Performance Based Transposition Algorithm and result of these recommendations shown through various extensive experiments.

Keywords—Recommender System, SEP, PBTA, Web Personalization, Web Logs, Category Based Recommendation

I. INTRODUCTION

The impact of the World Wide Web as a main source of information acquisition increasing dramatically. The existence of bulk of information, in combination with the dynamic and heterogeneous nature of the web, makes web site exploration a difficult process for the average user [1]. To address the requirement of effective web navigation, web sites provide personalized recommendations to the end users [2]. Recommender Systems (RSs) are software agents which give the suggestion for items to the user according to his/her preference [1][2][3][4]. These suggestions help user in decision-making processes such as what items to buy or what online news to read or what music to listen. Commercial websites use Recommender systems to suggest products to their customers [2]. Items are recommended to the customers on the basis of their demographic information and by analyzing their past buying behavior. Analyzing the past buying behavior of customers and their demographic information helps the site to adapt itself according to customer [1].

II. RELATED WORK

Recommender systems became major research area since the emergence of the first papers on collaborative filtering since the mid-1990s [7]. There has been lot of research done in the industry on developing new approaches to recommender systems. The interest in this field still remains high because it address a problem rich research area and because of lots of practical applications that assist users to handle information overloads and provide personalized recommendations, content and services to them. Examples of such applications include recommending books, CDs and other products at Amazon.com, movies by Movie Lens [8]. Recommender systems are broadly classified into the following three categories [1]:

A. Content-Based Recommendations

Collaborative Filtering (CF) systems gather user feedback in the form of ratings for items in a given area and identify similarities in rating behavior amongst numerous users in order to determine how to recommend an item. According to John S. Breese, David Heckerman, and Carl Kadie [9] CF methods can be further divided into neighborhood-based and model-based approaches. In neighborhood-based techniques, a subset of users is selected on the basis of similarity to the active user, and a weighted combination of their ratings is used to generating predictions for this user. Paul Resnick et al. [10] proved that the commonly used measure of similarity is the Pearson correlation coefficient between the ratings of the two users. When applied to millions of users and items, traditional neighborhood-based CF algorithms do not perform well due to the computational complexity of the search for similar users. As a alternative, Linden et al. [11] proposed item-to-item Collaborative Filtering. This approach matches user’s rated items to similar items.

B. Collaborative Recommendations

Content-based recommender system provides recommendations by comparing representations of content describing an item to representations of content that interests the user. These approaches are also referred as content-based filtering. Lots of research has been done in this area focusing on recommending items with associated textual content like web-pages, books, and movies.
C. Hybrid Approaches

In order to leverage the advantage of content-based and collaborative recommenders, various hybrid approaches that combine the two are proposed. Claypool et al. [13] merge the two predictions by utilizing an adaptive weighted average, where the weight of the collaborative component increases as the number of users accessing an item increases. Melville et al. [14] proposed a general framework for content-boosted Collaborative Filtering, where content-based predictions are applied to convert a sparse user ratings matrix into a full ratings matrix, and then a CF method is used to provide recommendations.

III. PROBLEM STATEMENT

From the above discussion it is clear that, Content-based methods can uniquely distinguish each user, but CF still has some key advantages over them. Firstly, CF can accurately predict in situations where there is not much content associated with items, or where it is difficult for computer to analyse the contents — ideas, opinions etc. Secondly a CF system is capable to provide serendipitous recommendations, i.e. it can recommend items that are not part of user’s profile but relevant to the user. Because of these reasons, CF systems have been used to build recommender systems in different domains. However they suffer from two fundamental problems:

A. Sparsity: Most users do not rate most items and hence the user-item rating matrix is typically very sparse [3]. Therefore the chances of finding a set of users with significantly similar ratings are usually low. This is the case when systems have a very high item-to-user ratio. This problem is also very common when the system is in the early stage of use.

B. The Cold-start Problem: New items and new users problems in recommender systems are jointly known as the coldstart problem [3][6]. In Collaborative Filtering systems, an item can be recommended only when some user has rated it before. This problem applies to both new items and obscure items. The new-item problem is also referred as the first-rater problem. On the other hand content based approaches do not depend on rating of user. In Content based approach, the content-based predictions of similar users is used to improve predictions for the active user. But, the new-user problem is difficult to tackle, since without any past knowledge of user choice impossible to find similar users or to design a content-based profile.

IV. SEMANTIC ENHANCED ARCHITECTURE FOR RECOMMENDER SYSTEM

Sparsity and Cold Start are two major problems of present recommender Systems. This problem can be resolved by implementing architectural of Semantic Enhanced Personalizer (SEP) for web personalization proposed by Sanjeev Kumar Sharma et al.[15]. This framework is consists of three modules of recommendation such as, original recommendation, semantic recommendation and category based recommendation as shown in Fig. 1.

A. Original Recommendation

The original recommendation consists of two components of recommendation such as context-based filtering and collaborative-based filtering. In the context-based filtering, recommendation of items will be based on extra contextual information related to item provided by the user (item-based recommendation) and the similarity of items will be computed using Pearson correlation coefficient. In this process, the system asks the user to input context information. The system then asks the user to type any free keywords. These keywords are input into the query engine, which use the context information to narrow down the search results. The free keywords are input to the Synonym Finder engine. Thereafter, different meanings of the entered keywords will be return to the user. Here objective is to discover the correct sense of the keyword used. All the outcome of the query parser and synonym finder senses are then shown to the user. At the same time a web service call is made to the Web Services to capture the reviews of the product shown to the user earlier. The parser slook for these meaning in the web database. If a match is found then the results will be shown in that category. The benefit of using this approach is that it assists to cover the disadvantages of the User based collaborative filtering engine like lack of user ratings, false ratings etc and deliver correct predictions to the users.

B. Semantic Recommendation

In this approach, all URLs will be stored in database on the basis of implicit feedback. That means if user satisfies the condition for implicit feedback then URL of that page will automatically stored in database. Thereafter Performance Based Transposition Algorithm (PBTA) will be implemented on to this database to produce the frequent URLs. The PBTA algorithm is improved version of Apriori algorithm [16]. Strong association rules, which satisfy the threshold value of minimum support and minimum confidence, will be generated using PBTA. Then the larger labeled clusters of similar and strong rules will be created using Efficient Semantic Clustering (ESC). The recommendation of items will be based on to the implicit ratings. It means, if a page visited by a user included in the ratings then this URL will be matched to the Left-Hand-Side (LHS) portion of association rule in the labeled cluster, if it matches then, URLs present at the Right-Hand-Side (RHS) portion of association rule will be recommended to the user.

C. Category Recommendation

In this component, all keywords of respective URL will be stored in database on the basis of implicit ratings. Thereafter, Performance Based Transposition Algorithm (PBTA) will be applied on to this database to generate the frequent keywords and strong association rules based on these keywords [15][16]. When user types the keywords in the query box for searching the item, these keywords will be matched with the Left-hand-Side portion of association rule. If these keywords are matching with one or more rule, then all URLs related to the keywords at the Right-hand–side portion of rules will be recommended to the user. Rather than searching for quality web pages, the users of this system would be
directly taken to quality web pages matching their personal interests and preferences. The theme behind the category-based recommendation is same as semantic recommendation that incorporate the content and usage data in the recommendation process.

![Fig. 1 Semantic Enhanced Personalizer](image)

The propose approach provides a solution of Cold-Start (new-item and new-user) using Context-based filtering. The new-user problem arises with content-based systems. In order to make accurate recommendations, the system must first learn the user’s preferences from the ratings that the user makes. The proposed approach has given the facility of context-based recommendation [15]. In which, additional information related to item provided by the user in the query box and recommendation of items will be based on synonyms or metadata related the contextual information. The other problem is new-item problem, in which new items are added regularly to recommender systems. Collaborative systems rely solely on users’ preferences to make recommendations. Therefore, until the new item is rated by a substantial number of users, the recommender system would not be able to recommend it. The proposed approach has given the facility of Context-Based Recommendation in which if new item is added with website, then this item will be recommended on the basis of contextual information provided by the user [15]. This system is also capable to recommend the items not yet rated by any user. Therefore, it doesn’t suffer from the first rater problem. This system has also the support of collaborative recommendation. Hence, the recommendation of items based on the other user with similar taste liked in the past.

V. PERFORMANCE BASED TRANSPOSITION ALGORITHM

Association Rule Mining is one of the promising techniques of data mining to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories [17]. The Apriori algorithm is also called the level-wise algorithm to find all of the frequent sets, which uses the downward closure property. The advantage of the algorithm is that before reading the database at every level, it prunes many of the sets which are unlikely to be frequent sets by using the Apriori property, which states that all nonempty subsets of frequent sets must also be frequent.

In Apriori algorithm, discovery of association rules require repeated passes over the entire database to determine the commonly occurring set of data items. Therefore, if the size of disk and database is large, then the rate of input/output (I/O) overhead to scan the entire database may be very high [16]. Performance Based Transposition Algorithm (PBTA), improves the Apriori algorithm for repeated scanning of large databases for frequent itemsets generation. In PBTA, transaction dataset will be used in the transposed form and the description of proposed algorithm is discussed in the following sub-sections.

A. Candidate Generation Algorithm

In the candidate generation algorithm, the frequent itemsets are discovered in k-1 passes. If k is the pass number, \( L_{k-1} \) is the set of all frequent (k-1) itemsets. \( C_k \) is the set of candidate sets of pass k and c denotes the candidate set. \( l_1, l_2, \ldots, l_k \) are the itemsets. The candidate generation procedure is as follows.

Procedure Gen_candidate_itemsets \((L_{k-1})\)

\[
C_k = \Phi
\]

for all itemsets \( l_1 \in L_{k-1} \), do

for all itemsets \( l_2 \in L_{k-1} \), do

if \( l_1[1] = l_2[1] \) \( ^ \wedge l_1[2] = l_2[2] \) \( ^ \wedge \ldots ^ \wedge l_1[k-1] < l_2[k-1] \)

© 2013, IJARCSSE All Rights Reserved
then \( c = l_1 \{1\}, l_1 \{2\} \ldots l_1 \{k-1\}, l_2 \{k-1\} \)
\( C_k = C_k \cup \{c\} \)

**B. Pruning Algorithm**
The pruning step eliminates some candidate sets which are not found to be frequent.
Procedure Prune\((C_k)\)

for all \( c \in C_k \)

for all \((k-1)\)-subsets \( d \) of \( c \) do

if \( d \in L_{k-1} \)

then \( C_k = C_k \setminus \{c\} \)

**C. PBTA Algorithm Description**
The PBTA uses candidate generation and pruning algorithms at every iteration. It moves from level 1 to level \( k \) or until no candidate set remains after pruning. The step-by-step procedure of PBTA algorithm is described as follows.

1. Transpose the transactional database
2. Read the database to count the support of \( C_1 \) to determine \( L_1 \) using sum of rows.
3. \( L_1= \) Frequent 1-itemsets and \( k:= 2 \)
4. While \((k-1) \neq NULL \) set do

Begin

\( C_k := \) Call Gen_candidate_itemsets \((L_{k-1})\)

Call Prune \((C_k)\)

for all itemsets \( i \in I \) do

Calculate the support values using dot-multiplication of array;

\( L_k := \) All candidates in \( C_k \) with a minimum support;

\( K:= k+1 \)

End

5. End of step-4

By comparing both Apriori and PBTA, we can conclude that Apriori algorithm requires multiple passes of the dataset to calculate support count for different itemsets. Therefore, in the case of Apriori, the record pointer moves the order of candidate item set \( \times \) no of records while in the case of PBTA algorithm, record pointer moves equal to only order of candidate itemsets [16].

**VI. PERFORMANCE ANALYSIS OF APRIORI AND PBTA USING WEKA**
Apriori and PBTA are tested on Weka which is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. It is also well suited for developing new machine learning schemes [18]. Weka is developed by the University of Waikato. In our paper we use Weka for comparing and analyzing performance of PBTA and Apriori Algorithm. We use msnbc.com dataset for comparing Apriori and PBTA. For processing the dataset on Weka, it is converted to Attribute Relation Format (ARFF).

The sample of dataset is shown below in Fig. 2.

![Sample of msnbc.com Dataset in ARFF Format](image-url)
VII. DESCRIPTION OF DATASET

For evaluating behavior of PBTA and Apriori Algorithm, we use the data set comes from Internet Information Server (IIS) logs for msnbc.com and news-related portions of msn.com for the entire day of September, 28, 1999 (Pacific Standard Time) [19]. Each sequence in the dataset corresponds to page views of a user during that twenty-four hour period. Each event in the sequence corresponds to a user's request for a page. Requests are not recorded at the finest level of detail—that is, at the level of URL, but rather, they are recorded at the level of page category (as determined by a site administrator). The categories are “frontpage”, “news”, “tech”, “local”, “opinion”, “on-air”, “misc”, “weather”, “health”, “living”, “business”, “sports”, “summary”, “bbs” (bulletin board service), “travel”, “msn-news”, and “msn-sports”. Any page requests served via a caching mechanism were not recorded in the server logs and, hence, not present in the data. The full Dataset consists of approximately one million users, with an average of 5.7 events per sequence. After bit of experimentation, we found the random sample of about 400,000 is adequate for process of generating strong association rules.

VIII. EXPERIMENTAL EVALUATION

The performance comparison of PBTA with Apriori Algorithm is presented in this Section. All the experiments are performed on 1.60 Ghz Pentium-Dual Core Laptop machine with 2 GB main memory, running on Window-7 operating system. We use Weka 3.7.9 Developer version code of Apriori for evaluation. The program for PBTA algorithm was developed in Java JDK1.7 environment. We use Eclipse Java IDE for coding and executing java programs. We report the experimental results on datasets of msnbc.com with 79K and 174K records, each having 17 columns. The dataset is converted into ARFF format for processing on Weka. The performances results of Apriori and PBTA are shown with Fig. 3 and Fig. 4 represented in the graphical form. The X-axis in these graphs represents the support threshold values while the Y-axis represents the response times (in seconds) of the algorithms being evaluated as shown.

In the first case, we have considered the transactional database with 79K records. We have compared the performance of Apriori with PBTA on the basis of response time as shown in Table I.

<table>
<thead>
<tr>
<th>Support Count (in %)</th>
<th>Apriori (in Sec)</th>
<th>PBTA (in Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>5.422</td>
<td>3.609</td>
</tr>
<tr>
<td>40</td>
<td>6.811</td>
<td>4.384</td>
</tr>
<tr>
<td>30</td>
<td>8.506</td>
<td>5.472</td>
</tr>
<tr>
<td>20</td>
<td>8.835</td>
<td>5.712</td>
</tr>
<tr>
<td>10</td>
<td>13.313</td>
<td>11.492</td>
</tr>
</tbody>
</table>

TABLE I Response Time Comparison of Algorithms with 79K Dataset

In the second case, we have considered the transactional database with 174K records. We have compared the performance of Apriori with PBTA on the basis of response time as shown in Table II.

In the second case, we have considered the transactional database with 174K records. We have compared the performance of Apriori with PBTA on the basis of response time as shown in Table II.
From the above Fig. 3 and Fig. 4, it is clear that while the support count is decreased then the response time taken by PBTA is much lesser than Apriori algorithm.

IX. IMPLEMENTATION OF CATEGORY BASED RECOMMENDATION MODULE USING PBTA

a. First the available data set is converted in Market-Basket dataset format for processing. That is each request of webpage (as category based recommender system is used for recommending WebPages) is given 1 value in data set. Missing values are treated as 0.

b. All the keywords of respective URLs are identified from the dataset and PBTA algorithm is applied on to this dataset to generate frequent keywords and strong association rule based on these keywords. Association rules generated by PBTA in weka are shown in Fig. 5.

Fig. 5. Association Rules generated using PBTA on Weka
c. Keywords related to Webpage URL are taken as input from the user as shown in Fig. 6.

d. The keyword entered by the user is matched with the left-hand portion of the association rule.
e. If these keywords are matching with one or more rules, then all URLs related to the keywords at the right-hand side portion of the rules will be recommended to the user. For example, if news, living→ frontpage is generated as a strong rule and the user types the keyword news in the textbox then news page opens and living and frontpage is recommended to the user as shown in Fig. 7.

X. CONCLUSIONS

All existing recommender systems use one or more basic techniques such as content-based, collaborative, demographic, utility-based etc. A study shows that all these techniques have some advantages and disadvantages. In this paper, we reviewed various drawbacks of the current recommendation methods and concluded that the architectural framework of SEP...
for recommendation of items overcomes the problem of existing recommender systems such as Cold-Start problem (new-user, new-items) and sparsity. We have implemented the last component of this framework i.e. category recommendation. This recommendation module implement PBTA algorithm for generating frequent keywords related to Webpage URL. The PBTA Algorithm is improved version of Apriori which is proved in this paper by testing PBTA and Apriori performance on msnbc.com dataset. We have shown the results of Category recommendations module through extensive experiments. In the future work, semantic recommendation module of SEP will be developed and importance of this recommendation technique will be shown through various experiments.

REFERENCES