**ONLINE RECOMMENDATIONS BY WEB USAGE MINING**

*A PROJECT REPORT*

**ABSTRACT**

The Internet is one of the fastest growing areas of intelligence gathering. During their navigation web users leave many records of their activity. This huge amount of data can be a useful source of knowledge. Sophisticated mining processes are needed for this knowledge to be extracted, understood and used. Web Usage Mining (WUM) systems are specifically designed to carry out this task by analyzing the data representing usage data about a particular Web Site. WUM can model user behavior and, therefore, to forecast their future movements. Online prediction is one web usage mining application. However, the accuracy of the prediction and classification in the current architecture of predicting users’ future requests systems cannot still satisfy users especially in Huge Web sites. To provide online prediction efficiently, we develop architecture for online recommendation for predicting in Web Usage Mining System .In this paper we propose architecture of on line recommendation in Web usage mining(OLRWMS) for enhancing accuracy of classification by interaction between classifications, evaluation, and current user activates and user profile in online phase of this architecture.

E-commerce data is rich and detailed compared to off-line commerce data. One of them is clickstream that means visitor’s path through a web site. Clickstream in the Internet shopping mall provides information essential to understanding shopping patterns or pre-purchase behaviors of customers such as what products they see, what products they add to the shopping cart, and what products they buy. Through analyzing such information (i.e., web usage mining), it is possible to make a more accurate analysis of customer’s interest or preference across all products than analyzing the purchase records only. Furthermore, mining association rules from clickstream provides rich and interesting relationships or associations among products, which are used in characterizing the appeal of individual products, compared to the conventional mining association rules from purchase records. Nevertheless, the existing researches could not afford to give a formal way for capturing individual customer’s preference or associations among products through web usage mining. In this study, we suggest methods to learn the customer preference and the product association from clickstream.

**Keywords**: Product recommendation; Personalization; Web usage mining; Clickstream.

**INTRODUCTION**

* 1. **OBJECTIVE**

With the explosive growth of knowledge available on the World Wide Web, which lacks an integrated structure or schema, it becomes much more difficult for users to access relevant information efficiently.

Meanwhile, the substantial increase in the number of websites presents a challenging task for webmasters to organize the contents of the websites to cater to the needs of users. Modeling and analyzing web navigation behavior is helpful for understand what information online users demand. Following that, the analyzed results can be seen as knowledge to be used in intelligent online applications, refining web site maps, and improving searching accuracy when seeking information. Nevertheless, an online navigation behavior grows each passing day, and thus extracting intelligently from it is a difficult issue. Web Mining has shown to be a viable technique to discover information “hidden” into Web-related data. In particular, Web Usage Mining (WUM) is the process of extracting knowledge from Web user’s access data by exploiting Data Mining (DM).It can be used for different purposes such as personalization, system improvement and site modification. Typically, the WUM prediction process is structured according to two components performed online and off-line with respect to the Web server activity. The off-line component is aimed at building the knowledge base by analyzing historical data, such as server access log files, that is then used in the online component. The main functions carried out by this component are Preprocessing, i.e. data cleaning and session identification, and Pattern Discovery, i.e. the application of DM techniques, like association rules, sequential patterns, clustering or classification. The online component is devoted to the generation of personalized content. On the basis of the knowledge extracted in the off-line component, it processes a request to the Web server by adding personalized content which can be expressed in several forms, such as links to pages, advertisements, and information relating to products or service estimated to be of interest for the current user. In the past, several WUM projects have been proposed to predict users’ preference and their navigation behavior, as well as many recent results improved separately the quality of the recommendations or the user profiling phase .existing collaborative filtering based recommendations suffer serious scalability problems.

Recent studies have suggested web usage mining as an enabler to overcome the problems associated with collaborative filtering since it will reduce the need for obtaining subjective user ratings or registration-based personal preference. E-commerce data is rich and detailed compared to off-line commerce data. One of them is clickstream that means visitor’s path through a web site. Clickstream in the Internet shopping mall provides information essential to understanding

shopping patterns or prepurchase behaviors of customers such as what products they see, what

what products they add to the shopping cart, and what products they buy. Through analyzing such information (i.e., web usage mining), it is possible to make a more accurate analysis of customer’s interest or preference across all products than analyzing the purchase records only. Furthermore, mining association rules from clickstream provides rich and interesting relationships or associations among products, which are used in characterizing the appeal of individual products, compared to the conventional mining association rules from purchase records. Nevertheless, the existing researches could not afford to give a formal way for capturing individual customer’s preference or associations among products through web usage mining. In this study, we suggest methods to learn the customer preference and the product association from clickstream.The quality of the recommendations has an important effect on the customer’s future shopping behavior. Poor recommendations can cause two types of characteristic errors: false negatives, which are products that are not recommended, though the customer would like them, and false positives, which are products that are recommended, though the customer does not like them. In an e-commerce environment, the most important errors to avoid are false positives, because these errors will lead to angry customers and thus they will be unlike to revisit the site. If we try to find customers who are likely to buy recommended products and recommend products to only them, that could be a solution to avoid the false positives of the poor recommendation.

**1.2 .PROPOSED WORK**

In our work, we propose a personalized recommendation methodology based on web usage mining. Further- more, decision tree induction is used to minimize recommendation errors by making recommendation only for customers who are likely to buy recommended products. For the implementation of the proposed methodology, a recommender system is also developed using intelligent agent and data warehousing technology. We begin by reviewing previous works related to our research in, the suggested recommendation methodology is explained with an illustra- tive example case. An agent based recommender system implemented for the evaluation is presented in. Finally, we summarize our contributions with suggestions for future research in which we simultaneously will be using:

-Traversal Pattern

-A second attribute using web analytics(Exit rate/Bounce rate)

In EC (Electronic Commerce) environment, how to find the association rules between purchasing items is very important. If we provider this kind of information to the web site manager, the performance of cross-selling should be improved. However, the web site managers may also pay attention to the navigation behaviors of customers. Association rule mining just purely considered the purchasing behaviors of customers. It cannot fully satisfy the managers’ requirements. Path traversal pattern mining is the technique that find most of the navigation behaviors of customers in the EC environment. The web site designer can use this information to improve the web site design, and to increase the web site performance .Besides, this information can also provide the navigation suggestions to customers. However, the web site manager considers not only the pure navigation behaviors but also the purchasing behaviors of customers. The path traversal pattern mining just purely considered purchasing behaviors of customers. It cannot fully satisfy the managers’ requirements. To overcome the disadvantages of the pure association rule mining and pure path traversal pattern mining, proposed a model that takes both the traveling patterns and purchasing behaviors of customers into consideration at the same time. We use an example to explain their method. Figure 1 is an example of web transaction of a customer. It shows the customer traverses from A to B (buys the item 1). Next, the customer goes to C, and then goes to D. Thereafter, the customer backs to C and then goes to E (buys item 3, 4). Finally, the customer goes back to C and buys the item 2 at that time used to discover the final results based on these web transaction records. When the customer has the backward behavior, a web transaction record is generated . Besides, if the last web page of a web transaction record does not purchase any items, it cannot be generated.

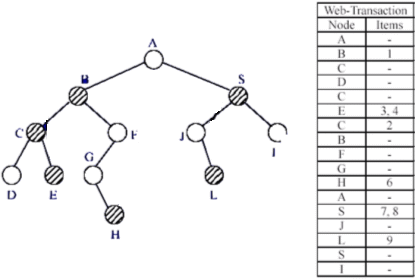


FIGURE 1

**CHAPTER 2**

**2.1. LATEST WORK**

Web usage mining is the process of applying data mining techniques to the discovery of behavior patterns based on web data, for various applications. In the advance of e- commerce, the importance of web usage mining grows larger than before. The overall process of web usage mining is generally divided into two main tasks; data preparation and pattern discovery. The data preparation tasks build a server session file where each session is a sequence of requests of different types made by single user during a single visit to a site. Cooley, Mobasher, and Srivastava (1999) presented a detailed description of data preparation methods for mining web browsing patterns. The pattern discovery tasks involve the discovery of association rules, sequential patterns, usage clusters, page clusters, user classifications or any other pattern discovery method (Mobasher et al., 2000a,b). Usage pattern extracted from web data can be applied to a wide range of applications such as web personalization, system improvement, site modifi- cation, business intelligence discovery, usage characteriz- ation, and so on (Srivastava, Cooley, Deshpande, & Tan,2000).

Our methodology recommends products based on web usage data as well as product purchase data and customer- related data. There have been several customer behavior models for e-commerce, which have different analysis purposes. Menasce´ , Almeida, Fonseca, and Mendes (1999) have presented a state transition graph, called Customer Behavior Model Graph (CBMG), that is used to describe the behavior of groups of customers who exhibit similar navigational patterns. VanderMeer, Dutta, and Datta (2000) have developed a user navigation model designed for supporting and tracking dynamic user behavior in online personalization. The model supports the notion of a product catalog, user navigation over this catalog and dynamic content delivery. Lee, Podlaseck, Schonberg, and Hoch (2001) have provided a detailed case study of the clickstream analysis from an online retail store. Among these models, a part of Lee et al.’s model is adopted to our research, because they focus the online retailer who is our consideration as well. In order to understand the effectiveness of web merchandising, they have analyzed e- shopper’s behavior according to the following four shopping steps: product impression, clickthrough, basket placement, and purchase. Micro-conversion rates (e.g., click-to-buy rate) used for measuring the effective of efforts in merchandising are computed for each adjacent pair of these steps. The study shows how the breakdown of clickstreams into sub-segments can highlight potential problems in merchandising. For example, a product may have many click-through but a low click-to-buy rate. Subsequent analysis may show that it has a high basket-to- buy rate, but a low click-to-basket rate. This analysis would allow merchandisers to begin to develop plans about how performance might be improved.

Given a set of transactions where each transaction is a set of items (itemset), an association rule implies the form X ) Y , where X and Y are item sets; X and Y are called the body and the head, respectively. The support for the association rule X ) Y is the percentage of transactions that contain both item set X and Y among all transactions. The confidence for the rule X ) Y is the percentage of transactions that contain item set Y among transaction that contain item set X. The support represents the usefulness of the discovered rule and the confidence represents certainty of the rule.

Association rule mining is the discovery of all association rules that are above a user-specified minimum support and minimum confidence. Apriori algorithm is one of the prevalent techniques used to find association rules (Agrawal, Imielinski, & Swami, 1993; Agrawal & Srikant, 1994). Apriori operates in two phases. In the first phase, all itemsets with minimum support ( frequent itemsets) are generated. This phase utilizes the downward closure property of support. In other words, if an itemset of size k is a frequent itemset, then all the itemsets below (k 2 1) size must also be frequent itemsets. Using this property, candidate itemsets of size k are generated from the set of frequent itemsets of size (k 2 1) by imposing the constraint that all subsets of size (k 2 1) of any candidate itemset must be present in the set of frequent itemsets of size (k 2 1). The second phase of the algorithm generates rules from the set of all frequent itemsets.

Association rule mining has been widely used from traditional business applications such as cross-marketing, attached mailing, catalog design, loss-leader analysis, store layout, and customer segmentation (Agrawal et al., 1993; Srikant & Agrawal, 1995), to e-business applications such as the renewal of web pages (Cooley et al., 1999) and web personalization (Mobasher et al., 2000a; Mulvenna, Anand, & Bu¨ chner, 2000).

In most Internet shopping malls, the product taxonomy is available. A product taxonomy T is practically represented as a tree that classifies a set of low-level products into a higher-level of a more general product. The leaves of the tree denote the product instances, stock keeping units (SKUs) in retail jargon, and non-leaf nodes denote product classes obtained by combining several lower-level nodes into one parent node. The root node labeled by All denotes the most general product class. Fig. 1 shows an example product taxonomy for a fashion Internet shopping mall, where Outerwear, Pants and Shirts are classified into Clothes, and so on.

A number called level can be assigned to each node in the product taxonomy. The level of the root node is zero, and the level of any other node is one plus the level of its parent. Please note that a higher-level product class has a smaller level number. The product taxonomy of Fig. 1 has four levels, referred to as levels 0 (for root), 1, 2, and 3.

Product taxonomies play an important role in the knowledge discovery process since they represent Inter- net shopping mall dependent knowledge and may affect the results. In many applications, strong association rules are more likely to exist at high levels of the product taxonomy but may likely repeat common knowledge. For example, the high-level association rule ‘80% of customers who buy clothes also buy footwear’ may be given to marketers of the fashion mall. On the other hand, low-level rules may be more interesting, but are difficult to find. For example, the low-level association rule ‘40% of customers who buy shirts also buy shoes’ could be mixed with many uninteresting rules. Therefore, it is important to mine association rules at the right level of the product taxonomy (Berry & Linoff, 1997; Han & Fu, 1995; Han & Kamber, 2001).

**2.2 IPA (Integrating Path Traversal Patterns) model**

Before illustrating our model, we firstly describe the definitions about support and confidence. The following shows an example of our mining results.

<ABCE: B{1} Ÿ E{4}>

This example explains that if the customers purchase item 1 at web page B, they usually buy item 4 at web page E and the traversal path is ABCE. We define the support of <ABCE: B{1},E{4}> is X/N. N denotes the number of total customers. X denotes the number of total customers who visit ABCE sequentially and purchase item 1 at web page B then purchase item 4 at web page E. We also define the support of <ABC: B{1}> is Y/N. N is the same as above. Y denotes the number of total customers who visit ABC sequentially and purchase item 1 at web page B. Based on the definition of support, we can define the confidence of <ABCE: B{1} Ÿ E{4}>. It is shown below.

Confidence(< ABCE : B{1} => E{4} >)

= Support(< ABCE : B{1}, E{4} >) = X Support(< ABC : B{1} >) Y

The confidence explains that customers traverse ABC and purchase item 1 at web page B, and the probability of purchasing item 4 at web page E is X/Y. Besides, a traversal sequence, e.g., ABC, is a set of web pages ordered by increasing traversal-time. The support for a traversal sequence is the number of customers who visit this sequence sequentially. If the support for a traversal sequence s satisfies the user-specified minimum support, then s is called a frequent traversal sequence. The length of a traversal sequence is the number of web pages in the sequence. A traversal sequence of length k is called a k-traversal sequence, and a frequent traversal sequence of length k is called a frequent k-traversal sequence. Similarly, a purchase sequence is a set of web pages ordered by increasing traversal-time, and at least one of these web pages has purchasing behaviors. The other definitions are the same as the traversal sequences. Based on these definitions, the main procedure is listed below.

**TABLE 2.Phases of Recommendation**

|  |  |
| --- | --- |
| Phase | Executing Step |
| Phase 1 | 1. List all candidate 1-travelsal sequences and candidate 1-purchase sequences. |

|  |  |
| --- | --- |
|  | 2. Scan database and calculate the support of candidate sequences. Find all frequent  1-traversal sequences and frequent  1-purchase sequences.  3. Join the frequent 1-traversal sequences and frequent 1-purchase sequences to the same set, and then join any two sequences to form a new one which just has a direct link in the web site structure. As a result, we can generate all candidate 2-traversal sequences and candidate 2-purchase sequences. |
| Phase K (K>=2) | 1. Scan database and calculate the supports of candidate K-traversal sequences and candidate K-purchase sequences separately. Then, find all frequent K- traversal sequences and frequent K-purchase sequences.  2. Put frequent K-traversal sequences and frequent K-purchases sequences to the same set and then join any two sequences to form a new one which just has a direct link in the web site structure to generate all candidate (K+1)-traversal sequences and (K+1)-purchase sequences. That is, for any two different frequent K-sequences  {r, S1 , S2 ,…, Sk -1 } and {S1 , S2 ,…, Sk -1 , *t*} , a  (K+1)-sequences {r,S1 , S2 ,…, Sk -1 , *t*  }  will be generated.  3. Execute steps 1 and 2 recursively, until no frequent sequences can be generated. |

In order to clearly explain our model, the following shows an example. The web site structure is shown in Figure 2 and the customer traversing and purchasing database is shown in Table 2. The notation represents that the customer purchases on that web page.

Suppose the minimum support is set to 2/6. First, we list all candidate 1-traversal sequences and candidate 1-purchase sequences (Phase 1, Step 1). In this example, all candidate 1-traversal sequences are {A, B, C, D, E, F} and all candidate 1-purchase.

**Table 3. Customer traversing and purchasing database**

|  |  |  |
| --- | --- | --- |
| ID | Access Path | Items Bought |
| 1 | BECAF’C | F{1} |
| 2 | DBAC’AE’ | C{2}, E{3} |
| 3 | BDAE |  |
| 4 | BDECAF’ C | F{1} |
| 5 | BAC’AE’ | C{2}, E{3} |
| 6 | DAC’ | C{2} |

sequences are {C’, E’, F’}. Next, we scan database and calculate the supports of these sequences. Then, we find all frequent 1-traversal sequences and frequent

1-purchase sequences (Phase 1, Step 2). In this step, the support count of C is 5. In order to count C, we add up the original support count of C, i.e., 2, and support count of C’, i.e., 3. Because the supports of all candidates are greater than the minimum support, all frequent 1-traversal sequences are {A, B, C, D, E, F} and all frequent 1-purchase sequences are {C’, E’, F’}.

Then, we put these sequences to the same set, i.e., {A, B, C, D, E, F, C’, E’, F’}, and then join any two sequences to form a new one which just has a direct link in the web site structure (Phase 1, Step 3). For example, if we join frequent 1-traversal sequence {A} and frequent 1-purchase sequence {C’}, then candidate 2-purchase sequence {AC’} will be generated. Because page A have a direct link to page C, we preserve this candidate 2-purchase sequence. According to this way, we can generate all candidate 2-traversal sequences, i.e., {AC, AE, AF, BA, BD, BE, CA, DA, DB, DE, EC, FC} and candidate 2-purchase sequences, i.e., {AC’, AE’, AF’, BE’, DE’, EC’, FC’, C’A, E’C, E’C’, F’C, F’C’}.

Next, we scan database and calculate the supports of all candidate K-traversal sequences and candidate K-purchase sequences to find all frequent sequences (Phase 2, Step 1). For instance, the support count of{AC} is 5 and {AC’} is 3. Based on the minimum support, we can generate all frequent 2-traversal sequences, i.e., {AC, AE, AF, BA, BD, BE, CA, DA, DE, EC, FC}, and frequent 2-purchase sequences, i.e.,{AC’, AE’, AF, BE’, C’A, F’C}. All of these sequences have the direct link on the web site structure.

As shown in Table 3, we consider not only forward information but also backward information, e.g., <ACAE: C{2} Ÿ E{3}>. Especially, our approach allows the noise exist in the transactions. For instance, <BAE: E{3}> is supported by customers 2 and 5, although BAE is not successive in the traversal paths of these two customers. BAE is counted in the counting procedure because there is a direct link between pages B and A, and also pages A and E. In other words, web pages between A and E can be seen as noises.

**TABLE 4.TRAVERSAL PATTERN**

|  |  |  |
| --- | --- | --- |
| Pattern | Suppor t | Confidenc e |
| <BECAF: F{4}> | 2/6 | 2/2 |
| <BACAE: C{2} Ÿ E{3}> | 2/6 | 2/2 |
| <ACAE: C{2} Ÿ E{3}> | 2/6 | 2/2 |
| <ECAF: F{4}> | 2/6 | 2/2 |
| <BAE: E{3}> | 2/6 | 2/5 |
| <CAF: F{4}> | 2/6 | 2/4 |
| <DAC: C{2}> | 2/6 | 2/4 |
| <CAE: C{2} Ÿ E{3}> | 2/6 | 2/2 |
| <AC: C{2}> | 3/6 | 3/6 |
|  |  |  |

|  |  |
| --- | --- |
| Traversal  Path | Purchases |
| ABCE | B{1}, C{2}, E{4}, E{3} |
| ABFGH | B{1}, H{6} |
| ASJL | S{7}, S{8}, L{9} |

**TABLE 5. TRAVERSAL RECORD**

The reason for cutting each web transaction into web transaction records is that they think the backward movement is not meaningful. Also, it can reduce the complexity of the algorithm. However, this assumption is not always true in all situations. In Figure 1, it is easy to see that the customer traverse to E and purchase items 3 and 4, and then back to C and purchase item 2 at that time. However, the web transaction records in Table 1 show that the customer buys the item 2 at C and then buys items 3 and 4 at E. Therefore, the order of traversing and purchasing is not correct after dividing. Thus, MTS algorithm cannot generate the correct results based on these web transaction records.

For this reason, this paper proposes a new algorithm, IPA (Integrating Path traversal patterns and Association rules), for mining web transaction patterns without dividing any web transactions. The IPA algorithm also takes both the traveling and purchasing behaviors of customers into consideration at the same time, and considers not only user forward traversal information but also backward information. Besides, web structure is also used in this algorithm to prune the unnecessary candidates. Especially, the IPA algorithm allows the noise exist in web transactions.

**2.3.AN EXAMPLE:**

The following shows an example of mining results.

**<ABCE: B{1} Ÿ E{4}>**

This example explains that if the customers purchase item 1 at web page B, they usually buy item 4 at web page E and the traversal path is ABCE. Now, we define the support of <ABCE: B{1},E{4}> is X/N. N denotes the number of total customers. X denotes the number of total customers who visit ABCE sequentially and purchase item 1 at web page B then purchase item 4 at web page E. We also define the support of <ABC: B{1}> is Y/N. N is the same as above. Y denotes the number of total customers who visit ABC sequentially and purchase item 1 at web page B. Based on the definition of support, we can define the confidence of <ABCE: B{1} Ÿ E{4}>. It is shown below.

Confidence( ABCE : B{1}! E{4} !)

Support( ABCE : B{1}, E{4} !) X

Support( ABC : B{1} !) Y

The confidence explains that customers traverse ABC and purchase item 1 at web page B, and the probability of purchasing item 4 at web page E is X/Y.

Besides, a traversal sequence, e.g., ABC, is a set of web pages ordered by increasing traversal-time. The support for a traversal sequence is the number of customers who visit this sequence sequentially divide by the number of total customers. If the support for a traversal sequence q satisfies the user-specified minimum support, then q is called a frequent traversal sequence. The length of a traversal sequence is the number of web pages in the sequence. A traversal sequence of length k is called a k-traversal sequence, and a frequent traversal sequence of length k is called a frequent k-traversal sequence. Before generating the frequent traversal sequences, we need to generate the candidate traversal sequences, and scan the database to count the support for each candidate traversal sequence. A candidate traversal sequence of length k is called a candidate k-traversal sequence. Similarly, a purchase sequence, e.g., AB’C, is a set of web pages ordered by increasing traversal time and at least one of the web pages B, has purchasing behaviors. The length of a purchase sequence is the number of web pages will be generated. Finally, the the sequence. A purchase sequence of length k is called a k-purchase sequence, and a frequent purchase sequence of length k is called a frequent k- purchase sequence.

**2.4. IPA Algorithm**

Based on the definitions described in Section 2, the IPA (Integrating Path traversal patterns and Association rules) algorithm is listed below. The input of the IPA algorithm is the customer traversing and purchasing database, i.e., TDB, the user-specified minimum support for traversal sequence, i.e., s, and the user-specified minimum support for purchase sequence, i.e., s’. Because most of the users do not purchase any items in access paths, we usually set s’ to s \* (t / n). The t denotes the total number of web transactions which have purchasing behaviors and n denotes the total number of web transactions. In IPA algorithm, CTS .k denotes the candidate k-traversal sequence, and CPS. k denotes the candidate k-purchase sequence. The FTS k denotes the frequent k-traversal sequence, and FPS. k denotes the frequent k purchase sequence.

**ALGORITHM**

**Begin**

**(FTS.1, FPS.1, M) = Count1(CTS.1, CPS.1, s, s’, TDB);**

**(FTS.2, FPS.2) = Scan(M, s, s’);**

**(CTS.3, CPS.3) = CandidateGeneration(FTS.2, FPS.2);**

**k = 3;**

**While(CPS.k z I)**

**begin**

**(FTS.k, FPS.k) = CountK(CTS.k, CPS.k, s, s’, TDB);**

**TDB = TrimDB(FTS.k, FPS.k, s, TDB);**

**(CTS.(k+1), CPS.(k+1)) = CandidateGeneration(FTS.k, FPS.k);**

**k++;**

**end**

The IPA algorithm uses the Count1 module to count the supports of CTS.1 and CPS.1 based on the TDB, and output FTS.1 and FPS.1 based on s and s’. The Count1 module also constructs a two-dimensional matrix M for counting the possible CTS.2 and CPS.2. The Scan module is used to output FTS.2 and FPS.2 based on M, s and s’. The CandidateGeneration module then firstly put FTS.(k-1) and FPS.(k-1) into a set A. For any two different frequent (k-1)-sequences in A, i.e., CandidateGeneration module checks the subsequences of all generated k-sequences. If a subsequence, which has a direct link in the web site structure for any two consecutive web pages of subsequence, is not frequent, the k-sequence will be eliminated.

**2.5. ANALYSIS OF OBSERVATIONS**

When CPS.k is not empty, the IPA algorithm will generate longer patterns. FTS.k and FPS.k are generated by the CountK module based on CTS.k, CPS.k, s, s’, and TDB. The TrimDB module then is used to remove redundant records from TDB.

In order to clearly explain our algorithm, the following shows an example. The web site structure is shown in Figure 2, and the customer traversing and purchasing database (TDB) is shown in Table 2.

Suppose the minimum support for traversal sequence is set to 3/6. Then, the minimum support for purchase sequence is usually set to 3/6 × 5/6 (the minimum support count is 2).

In this example, CTS.1 is {A, B, C, D, E, F} and CPS.1 is {C', E', F'}. Next, we count the supports of CTS.1 and CPS.1. After the Count1 module, we find that FTS.1 and FPS.1 are the same as CTS.1 and CPS.1, respectively. In Count1 module, the support count of C is added up the original support count of C, i.e., 2, and support count of C', i.e., 3. This is because C’ includes traveling and purchasing behaviors.

At the same time, we construct a two-dimensional matrix M in Count1 module for counting the possible CTS.2 and CPS.2. The results are listed below.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | C’ | E’ | F’ |
| A | 0 | 0 | 5 | 0 | 3 | 2 | 3 | 2 | 2 |
| B | 5 | 0 | 0 | 2 | 5 | 0 | 0 | 2 | 0 |
| C | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D | 4 | 1 | 0 | 0 | 3 | 0 | 0 | 1 | 0 |
| E | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| C’ | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| E’ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F’ | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 |

**Matrix M denoting Click Stream**

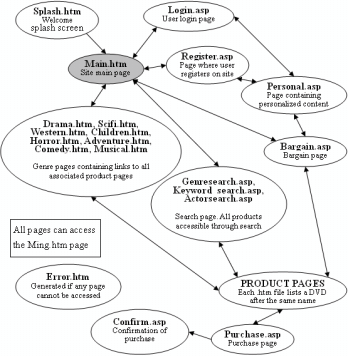
For the first access path, i.e., BECAF’C, it originally supports sequences BE, BC, BA, BF, BF’, EC, EA, EF, EF’, CA, CF, CF’, AF, AF’, AC, FC and F’C. Because BC, BF, BF’, EA, EF, EF’, CF and CF’ do not have a direct link between web pages, the first access path only supports BE, BA, EC, CA, AF, AF’, AC, FC and F’C. That is, only these 9 sequences are counted in M. According to M and two minimum supports, i.e., s and s’, we can FTS.2 and FPS.2. In this example, FTS is {AC, AE, BA, BE, CA, DA, DE} and FPS.2 is {AC', AE', AF', BE', C'A, FC'}. FPS.2 into the same set, i.e., {AC, AE, BA, BE, CA, DA, DE, AC', AE', AF', BE', C'A, FC'}. For any two different frequent 2-sequences, i.e., ^r, S1` and ^S1, t`, a3-sequences ^r, S1 , will be generated. According to this way, we can generate the following 3-sequences, i.e., {ACA, BAC, BAE, CAC, CAE, DAC, DAE, AC'A, AF'C, BAC', BAE', BAF', CAC', CAE', CAF', DAC', DAF', C'AC, C'AE, C'AC', C'AE', C'AF', F'CA}. After we generate these 3-sequences, we check each subsequence of these 3-sequences. If a subsequence, which has a direct link in the web site structure for any two consecutive web pages of subsequence, is not frequent, the sequence will be eliminated. For example, DAE' will be eliminated. This is because D and E have a direct link in the web site structure, and DE' is not frequent in FPS.2. After this process, we output FTS.3 and FPS.3. In this example, FTS.3 is {ACA, BAC, BAE, CAC, CAE, DAC, DAE} and FPS.3 is {AC'A, AF'C, BAC', BAE', BAF', CAC', CAE', CAF', DAC', DAF', C'AC, C'AE, C'AC', C'AE', C'AF', F'CA}. After we find FTS.3 and FPS.3, we calculate the possible generation length for each transaction to trim TDB. It is listed in Table 4. In this example, we trim two records, i.e., IDs 3 and 6, because the possible generation lengths for these two IDs are all less than 4.

|  |  |  |
| --- | --- | --- |
| ID | Access Paths | Possible Generation Length |
| 1 | BECAF’C | 4 |
| 2 | DBAC’AE’ | 5 |
| 3 | BDAE | 3 |
| 4 | BDECAF’C | 4 |
| 5 | BAC’AE’ | 5 |
| 6 | DAC’ | 3 |

**Table 7:Possible Generation lengths**

When k is equal to 5, we can generate all frequent sequences. Suppose the minimum confidence is set to 1/2. The final mining results are listed below.

|  |  |  |
| --- | --- | --- |
| Patterns | Support | Confidence |
| <BECAF: F{4}> | 2/6 | 2/2 |
| <BACAE: C{2} Ÿ E{3}> | 2/6 | 2/2 |
| <ACAE: C{2} Ÿ E{3}> | 2/6 | 2/2 |
| <ECAF: F{4}> | 2/6 | 2/2 |
| <BAE: E{3}> | 2/6 | 2/5 |
| <CAF: F{4}> | 2/6 | 2/4 |
| <DAC: C{2}> | 2/6 | 2/4 |
| <CAE: C{2} Ÿ E{3}> | 2/6 | 2/2 |
| <AC: C{2}> | 3/6 | 3/6 |



**FIGURE : WEBSITE STRUCTURE**

As shown in Table 5, we consider not only forward information but also backward information, and we use web site structure to prune unnecessary candidates. Especially, our algorithm allows the noise exist in the transactions. For example, suppose that a user traverses AXBC and the web site structure has a direct link between A and B. Although AB is not successive in the access path, AB is counted in the counting procedure. In other words, X can be seen as a noise, hence we can ignore it.

**2.6.TECHNOLOGIES USED**

**7.1IMPLEMENTATION:**

**7.1.1.Using Servlet in Java.**

**Configure the development environment**

To perform the steps outlined in this article, you will need to set up your development environment as described here:

1. **Download Apache Mahout**
   1. Visit the [the Apache Mahout web site](https://cwiki.apache.org/confluence/display/MAHOUT/Downloads) and download the latest version of Apache Mahout. As the time of this writing, the latest version was 0.5 (mahout-distribution-0.5.zip).
   2. Extract the contents of the archive to a known location. These content files will be referenced later.
2. **Create the Java EE application project**
   1. Start Rational Application Developer for WebSphere Software V8.0.3.
   2. Switch to the Java EE perspective.
   3. Select **File > New > Enterprise Application Project**.
   4. For Project Name, enter RecommenderApp.
   5. Ensure the target runtime is set to **WebSphere Application Server v8.0** and click **Next**.
   6. On the next panel, click **New module...** and from the popup, select **Web module only** and set its name to RecommenderWeb.
   7. Click **OK** and then **Finish**.
3. **Create and populate database with sample data**

Apache Derby is a Java-based database that uses a file store for storage. Apache Derby is used in this example because it is included with Rational Application Developer.

* 1. Select **Window > Show View > Data Source Explorer**.
  2. Right click on **Database Connections** and select **New ...**
  3. For JDBC driver, select **Derby 10.5 – Embedded JDBC Driver Default**.
  4. Since Derby stores databases in the file system, you need to specify where it will reside. For Database location, enter the path and name to use for the database. For this sample, PREFERENCES is used as the as the database name.
  5. Leave the username and password fields blank and click **Finish**.

Next, you will define your data model using a script (Listing 1). The script first creates a schema called PREFERENCES and a table called taste\_preferences. This table holds all the ratings users make about each item. It contains four columns: user\_id, item\_id, preferences and timestamp. Each tuple in the table represents that user user\_id has ranked item item\_id and given it a rating of preference

**Configure the Apache Mahout libraries**

In order to develop the recommender code, you need to import the required Apache Mahout libraries into your enterprise application. (If you are going to share the Apache Mahout libraries among multiple enterprise projects in your environment, then the recommended method would be to configure a shared library.)

Expand **EAR Projects**, right click on **RecommenderApp**, and select **Import > Import ...**

1. Select **General > File System** and click **Next**.
2. In From directory, browse to the location where you extracted the Apache Mahout files and click **OK**.
3. Select the files listed in the table below from the import dialog:

| **Library name** | **Location** |
| --- | --- |
| mahout-core-0.5.jar | mahout-distribution-0.5/ |
| mahout-core-0.5-job.jar | mahout-distribution-0.5/ |
| slf4j-jcl-1.6.0.jar | mahout-distribution-0.5/lib |

1. Click **Finish**.
2. Now add these references to the Web application so you can define your class path for compilation. Right click on the **RecommenderWeb** project and select **Properties**.
3. Select **Java Build Path** and click **Add JARs ...**
4. From the popup dialog, expand **RecommenderApp**, select **mahout-core-0.5.jar** and **mahout-core-0.5-job.jar** and click **OK** (Figure 5

Next, you will create the servlet that will handle the recommendation engine code from Apache Mahout.

**Create a servlet class**

**DATABASE-SQL(accessed using JDBC).**

**2.7.CONCLUSION**

We have presented a methodology for personalized recommendations in e-commerce and developed a recom- mender system implementing the methodology. The characteristics of the suggested methodology are as follows. First, the customer preference and the product association are automatically learned from clickstream (web usage data), unlike other recommendation methodologies which learn them from purchase records only. Second, in order to avoid the poor recommendations that will lead to disappoint customers, customers who are likely to buy recommended products are selected using decision tree induction. Third, the explicit participation of the marketers and the formal usage of background knowledge such as the product taxonomy are also introduced in the recommendation process. Finally, we devise measures to choose highly merited products among the candidate recommendable products.As further work, it will be interesting to compare our suggested methodology with a standard collaborative filtering based methodology in the aspect of recommendation performance. And it will also be an interesting research area to conduct a real marketing campaign to customers using this methodology and to evaluate the performance.

**CHAPTER 3**

**REFERENCES**

1. **Scienstein: A Research Paper Recommender System**

Bela Gipp1, Jöran Beel1, Christian Hentschel2

1 Otto-von-Guericke University, Dept. of Computer Science, Magdeburg, Germany

2 Fraunhofer Institute for Telecommunications, Berlin, Germany

Bela@Gipp.com, Joeran@Beel.org, [christian.hentschel@hhi.fraunhofer.de](mailto:christian.hentschel@hhi.fraunhofer.de)

1. **A Survey of Trust and Reputation Systems for**

**Online Service Provision**

Audun Jøsang a;1;4 Roslan Ismail b;2 Colin Boyd a;3

aInformation Security Research Centre

Queensland University of Technology

Brisbane, Australia

bCollege of Information Technology

Universiti Tenaga Nasional (UNITEN)

Malaysia

**3**.**Capturing knowledge of user preferences: ontologies in recommender systems**

Stuart E. Middleton, David C. De Roure and Nigel R. Shadbolt

Department of Electronics and Computer Science

University of Southampton

Southampton, S017 1BJ, UK

Email : {sem99r,dder,nrs}@ecs.soton.ac.uk

**4. A Web Service Recommender System Using Enhanced Syntactical Matching**

M. Brian Blake and Michael F. Nowlan

Department of Computer Science

Georgetown University

Washington, DC

{mb7, mfn3} @georgetown.edu

**5. Effects of Session Representation Models on the Performance of Web Recommender Systems**

G¨ul Nildem Demir Murat G¨oksedef A. S¸ima Etaner-Uyar

Istanbul Technical University

ndemir, goksedef, [etaner@itu.edu.tr](mailto:etaner@itu.edu.tr)

**6. A New Approach For On Line Recommender System in Web Usage Mining**

Subhash K.Shinde, Dr.U.V.Kulkarni

Department of Computer Engineering, Bharati Vidyapeeth Collge of Engineering,Navi Mumbai, India.

Department of Computer Engineering , S.G.G.S College of Engineering & Technology, Nanded, India.

E-mail: skshinde@rediffmail.com, [kulkarniuv@yahoo.com](mailto:kulkarniuv@yahoo.com)

**7. The Impact of Contextual Information on the Accuracy of Existing**

**Recommender Systems forWeb Personalization**

Marcos A. Domingues

Faculty of Science

University of Porto, Portugal

LIAAD-INESC Porto L.A.

marcos@liaad.up.pt

Al´ıpio M. Jorge

Faculty of Economics

University of Porto, Portugal

LIAAD-INESC Porto L.A.

amjorge@fep.up.pt

Carlos Soares

Faculty of Economics

University of Porto, Portugal

LIAAD-INESC Porto L.A.

[csoares@fep.up.pt](mailto:csoares@fep.up.pt)

8. **NetPersonal: A Recommender System to Personalize the Environment of Web User**

Bhawesh Kumar Thakur Syed Qamar Abbas Ashish Kumar Trivedi

Computer Sci. & Engg. Computer Sci. & Engg. Computer Sci. & Engg.

S.C.E.R.T, U.P.T.U. Integral University, I.E.M.,U.P.T.U,

Lucknow, India Lucknow, India Lucknow, India

bhawesh\_k thakur@yahoo.co.in qrat\_abbas@yahoo.com [ashishmtech@gmail.com](mailto:ashishmtech@gmail.com)

1. **An Efficient Algorithm for Web Recommendation Systems.**
2. **WSRec: A Collaborative Filtering Based Web Service Recommender System**

Zibin Zheng, Hao Ma, Michael R. Lyu, and Irwin King

Department of Computer Science and Engineering

The Chinese University of Hong Kong

Shatin, N.T., Hong Kong, China

{zbzheng, hma, lyu, [king}@cse.cuhk.edu.hk](mailto:king%7d@cse.cuhk.edu.hk)

1. **A Web Data Mining Framework for E-commerce Recommender Systems**

Jinhua Sun

Department of Computer Science and Technology

Xiamen University of Technology, XMUT

Xiamen, China

E-mail: jhsun@xmut.edu.cn

Yanqi Xie

Department of Computer Science and Technology

Xiamen University of Technology, XMUT

Xiamen, China

E-mail: [yqxie@xmut.edu.cn](mailto:yqxie@xmut.edu.cn)

1. **A Hybrid Recommender System Combining Web**

**Page Clustering with Web Usage Mining**

Liang Wei

Department of Management Science and Engineering

School of Management, University of Jinan

Jinan China

sm\_liangw@ujn.edu.cn

Zhao Shu-hai

Department of Management Science and Engineering

School of Management, University of Jinan

Jinan China

[sm\_zhaosh@ujn.edu.cn](mailto:sm_zhaosh@ujn.edu.cn)

1. **A Time Influence Analysis for Web Page Recommender System**

Chen-Chung Chi, Nai-Lung Tsao, Chin-Hwa Kuo

Department of Computer Science and Information Engineering,

Tamkang University

ryanjih@mail.tku.edu.tw, beaktsao@gmail.com, [chkuo@mail.tku.edu.tw](mailto:chkuo@mail.tku.edu.tw)

1. **A Dynamic Web Recommender System Based on Cellular Learning Automata**

Mojdeh Talabeigi

Department of Computer Engineering

Islamic Azad University, Qazvin Branch

Qazvin, Iran

Mojde.talabeigi@gmail.com

Rana Forsati

Department of Computer Engineering,

Islamic Azad University, Karaj Branch

Karaj, Iran

forsati@kiau.ac.ir

Mohammad Reza Meybodi

Department of Computer Engineering,

Amirkabir University of Technology

Tehran, Iran

mmeybodi@aut.ac.ir