International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:10, 2008

A Semantic Recommendation Procedure for Electronic Product Catalog

Hadi Khosravi Farsani, and Mohammadali Nematbakhsh

***Abstract*—**To overcome the product overload of Internet shoppers, we introduce a semantic recommendation procedure which is more efficient when applied to Internet shopping malls. The suggested procedure recommends the semantic products to the customers and is originally based on Web usage mining, product classification, association rule mining, and frequently purchasing. We applied the procedure to the data set of MovieLens Company for performance evaluation, and some experimental results are provided. The experimental results have shown superior performance in terms of coverage and precision.

***Keywords*—**Personalization, Recommendation, OWL Ontology, Electronic Catalogs, Classification.

I. Introduction

GROWING demands for the world-wide-web and the emergence of e-commerce have pushed designers to develop recommender systems [10]. Recommender systems are personalized information filtering technology used to either predict whether a particular user will like a particular item (prediction problem) or to identify a set of N items that will be of more interest to a certain user(top-N-recommendation problem). In recent years, recommender systems have been used in a number of different applications such as recommending products a customer will most likely buy; movies, TV programs, or music a user will find enjoyable; identifying web pages that will be of interest; or even suggesting alternate ways of searching for information [9].

The quality of the recommendations has an important effect on the customer's future shopping behavior. Unqualified recommendations may cause two types of characteristic errors*: false negative and false positive*. The former refers the product that is not recommended, although the customer would prefer it, and the latter refers the product that is recommended, although the customer would not prefer. In an e-commerce environment, the most important error that must be avoided is the false positive, since this error will lead to angry customers and thus they will be unlikely to revisit the site [4].

Manuscript received September 18, 2006. This work was supported in part by Department of Computer Engineering, University of Isfahan, Iran.

H. Khosravi Farsani, Department of Computer Engineering, University of Isfahan (hadi_farsani@yahoo.com).

M. NematBakhsh, Department of Computer Engineering, University of Isfahan (nematbakhsh@comp.ui.ac.ir).

To date, most personalization systems have fallen into three major categories; manual decision rule systems, collaborative filtering systems, and content-based filtering systems [7]. Manual decision rule systems enforce administrator to specify rules based on static profiles or session histories. The rules are used to affect the content served to a particular user. Collaborative filtering systems take information in the form of user ratings or preferences, afterwards try to discover similarity between items or users and accordingly, recommend the product. Content-based filtering approach relies on content similarity of items to personal profiles obtained explicitly or implicitly from users [11].

Two approaches have been developed to realize the collaborative filtering-based systems. The first approach, refereed to as *user-based*, relies on the fact that each person belongs to a larger group of similarly behaving individuals. As a result, the items frequently purchased by various members of the group can be used to form the basis for recommended items [1, 2, and 15]. The second approach, known as *model-based,* analyzes historical information to identify relations between different items, e.g., the purchase of an item often leads to the purchase of another item, and then uses theses relations to determine the recommended items. User-based collaborative filtering approaches tend to produce systems that lead to higher quality recommendations in comparison with model-base schema [14]. But the complexity of computing each recommendation grows linearly with the number of users and items. These systems also have 'new item' problem which means that new items do not recommended until they are bought or visited by other customers.

Designing an effective recommender system must tackle into a number of challenges. One of them is the modeling and presenting recommendation items. Modeling of customers' behavior is another challenge. Till now, there is no comprehensive model for customers' behavior, but they can almost present customers' behavior. The third challenge is the recommender algorithm and its quality. The last challenge is evolution of recommender system. There is not any criterion accepted by all for comparing recommendation systems [16, 17].

The new generations of Web personalization tools are attempting to incorporate techniques for pattern discovery from Web usage data. Web usage mining systems run any number of data mining algorithms on usage data which is gathered from one or more Web sites in order to discover user profiles. The increasing focus on Web usage data is due to several factors. The input is not a subjective description of the

users by the users themselves, and thus is not prone to biases. The profiles are dynamically obtained from user patterns, and thus the system performance does not degrade over time as the profiles grow. Furthermore, using only the content similarity as a way to obtain aggregate profiles may result in missing important semantic relationships between Web objects. Thus, Web usage mining can reduce the necessity for obtaining subjective user ratings or registration- based personal preferences [3, 5, 6, and 8].

Principal elements of Web personalization include the modeling of Web objects (for example, products or pages) and subjects (customers), Categorization of objects and subjects, matching between and across objects and/or subjects, and determination of the set of actions to be recommended for personalization. In this paper we will employ all of these elements for web personalization. This paper has been organized as follows. In section 2 we model and classify products. Section 3 contains modeling of the customers. Recommendation procedure will be discussed in section 4. The experimental results are presented in section 5 and finally the paper will be concluded in section 6.

II. Product Class by OWL

A *(product)* class is a set of products. Taking P as the set of all products in consideration, this class is a subset of P. A class can be defined by enumerating its members. Each class has several attributes. From among of these attributes, SubClassOf, WordNet, UNSPSC, ECLS@SS are very important. SubClassOf attribute is used for classification hierarchy. We use WordNet attribute for the meaning and purpose of the class. UNSPSC and ECL@SS are used for mapping between classes of these classification system and standard classification systems. We define product in consideration as below

Definition: A *product set* P_s over a set of products **P** is a 3-tuple<P, PC, PC_1>, where **P** is the set of all products (in consideration), **PC** = {PC_1, PC_2…,PC_x} is the set of all product classes that each class is in the form of {(A,V), R}.

P= {P_1, P_2….P_m} is the set of all products. PC_1 is the root product class. PC_i=< {(A_1, V_1), (A_2, V_2)…. (A_n, V_n)}, R>. A_1, A_2, A_3 and A_4 attributes are used as SubClassOf, WordNet, UNSPSC and ECL@SS property.

Using ontology and ontology languages for product classification system is the best choice. According to the studies that have been performed, OWL is selected as the best language from among the current ontology languages [25, 13]. The abilities of OWL ontology language consider the requirements of electronic catalog [26]. In this paper, we will present electronic catalog using OWL ontology language.

We must use ontology primitives for modeling e-catalogs. Each of current concepts in e-catalogs is mapped equivalently in OWL ontology language. Classes of products will be presented by *class* in ontology language [12, 8]. Product specifications will be described using object attributes. In this model, class hierarchy in electronic catalogs will be implemented with *ISA* or *SubClassOf* primitive. And finally, restrictions on the class or attribute are modeled using *Domain*, *Range*, etc.

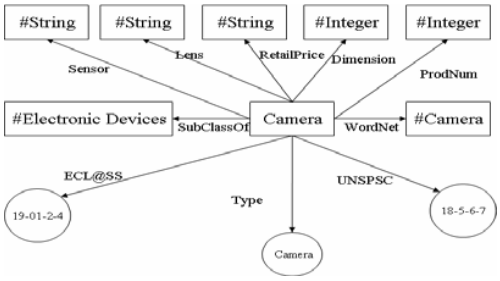

Fig. 1 Camera Class Definition in RDF

Product Information is set of attributes and values. For example, 'Camera' product has color attribute that has 'silvery' value. This data modeling was supported by RDF. RDF data modeling is A (O, V) where 'O' object has 'A' attribute that its value is 'V'. Camera class is a subclass of electronic devices which its type is 'camera'. It has sensor, prodNum, lens, dimension and retailPrice attributes. The camera class with its corresponding attributes is shown in Fig. 1.

In this model, each class also has UNSPSC, ECL@SS and other attributes. These attributes help customers to compare products among electronic markets. This class and one of its attributes are modeled as below.

```
<OWL: Class RDF: ID= 'Camera'>
<RDFS: SubClassOf RDF: resource =' #Electronic   Device '   />
</OWL: Class>

<RDF: Property  RDF: ID='ProdNum' >
          <rdfs:Domain    rdf:Resource = '#Camera' />
          <rdfs:Range    rdf:Resource='#integer'    />
</RDF: Property>
```

'Film Camera' class is a subclass of 'camera' class. This class inherits all of the attributes of super class and has other attributes such as 'filmSize', 'loading', and 'filmSpeed'. As you can see in Fig. 2, the value of sensor attribute of this class is 'film'. OWL ontology language will provide this restriction by using <OWL: restriction> tag. Implementation of this class in OWL ontology language is similar to 'camera' class.

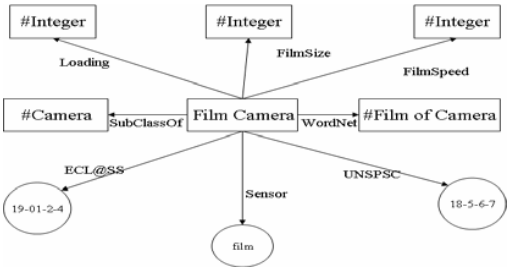

Fig. 2 Film Camera Class Definition

```
<OWL: Class   RDF: ID ='Film Camera'>
   <RDFS: SubClassOf   RDF: Resource='#Camera'>
     <OWL: Restriction >
```

```
        <OWL: onProperty   RDF:Resource='#sensor' />
         <OWL: HasValues> film </OWL: HasValuse>
      </OWL: Restriction >
    </RDFS: SubClassOf >
</OWL: Class>
```

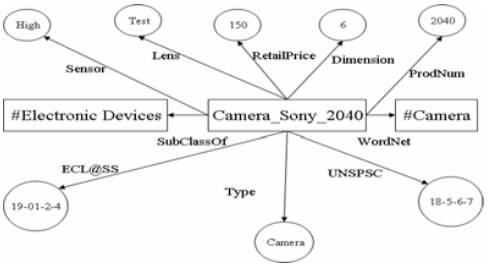

Fig. 3 Instance of Camera Class

'High end digital camera' is subclass of both 'digital camera' and 'high end camera'. We can present this situation using <OWL: Unionof> tag in OWL. If a class is defined by this tag, it inherits all of the attributes from super classes. Also, <OWL: Intersectionof> tag is used for intersection of two classes.

```
<OWL: Class  RDF: ID = 'High End Digital Camera'>
   <OWL:Unionof RDF: ParseType='collection'
      <OWL: Class RDF: about='#Digital Camera' />
      <OWL: Class RDF: about='#High End Camera' />
   </OWL:Unionof>
</OWL: Class>
```

We can define class instances in OWL language (See fig 3). For example, 'Camera-Sony-2040' is instance of 'camera' class.

```
< Camera RDF: ID = ' Camera_Sony_2040' >
          <Pro_num> 2040 </Prod_num>
          <manufacture rdf:resource= '# Sony Company'/>
          ...
</Camera>
```

We can improve this model using other primitives of OWL. <OWL: disjointwith> tag is used for defining independent classes. We will use <OWL: equivalentclass> tag for two class that are the same. In OWL, the maximum and minimum of an attribute will be determined using <OWL: maxcardinality> and <OWL:mincardinality> tags. Finally, we have a classification hierarchy for products after using these primitives from OWL language as you can see in Fig. 4.

III. Customer Class by OWL

The proposed system classifies customers using the same model which is used for product classification. It has defined a number of attributes for each customer class such as name, description, genre, and so on.

Definition: A *customer set* C_s over a set of Customers C is a 3-tuple<C, CC, CC_1>, where C is the set of all products (in consideration), **C**C = {CC_1, CC_2…, CC_y} is the set of all customer classes that each class is in the form of {(A, V), R}.

C= {C_1, C_2….C_z} is the set of all customers. CC_1 is the root cusomer class. CC_i=< {(A_1, V_1), (A_2, V_2)…. (A_n, V_n)}, R>. A_1, A_2, A_3 and A_4 attributes are used for SubClassOf, WordNet, UNSPSC and ECL@SS property.

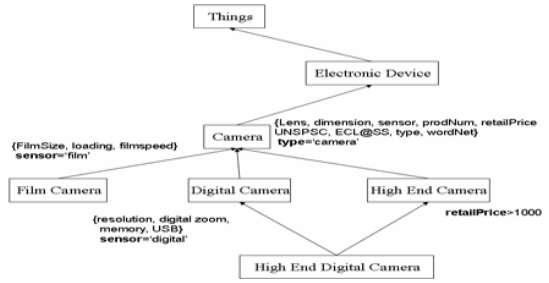

Fig. 4 Class Hierarchy of Products

IV. Recommendation Procedure

This paper suggests a methodology for personalized recommendations in an e-commerce environment. The methodology consists of five phases as shown in Fig. 5. The recommendation problem should be initiated by classifying the products and customers to improve the analyze process of similar customers and products. In phase II, active customers will be selected by considering a number of previous recommendations. The system does not recommend to a customer if the number of earlier recommendations to his/her has not exceeded a dedicated amount of threshold. Class of customer has been determined in phase III. The next two steps contribute in the process of performing recommendation, utilizing the proposed rating matrix.

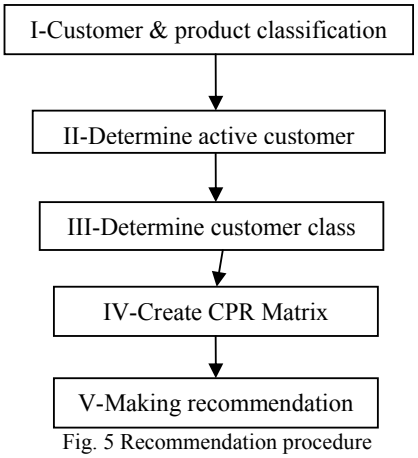

Fig. 5 Recommendation procedure

A. Identify Active Customers

Making recommendation, only the customers who are likely to buy the recommended products are considered. Since, the false positives of a poor recommendation will be avoided. This phase performs the role of selecting such customers based on previous Top-N Recommendations.

The recommendation system maintains previous Top-N Recommendation in the profile of each customer. The profile of one customer includes two parts. First part is used for

presenting personal information such as instanceOf, numRecommend, name, age, location and etc. InstanceOf is used to indicate that to which class the customer belongs. A number of previous recommendations are maintained by numRecommend.

If the number of previous recommendations of one customer exceeded to maxRecommend (which is maintained in customer class), the system deactivates this customer and in future do not recommend any product to her/his. The customer will be active when he/she purchases some of products where the numRecommend variable will be reset to zero to let the system re-recommends the products to this customer.

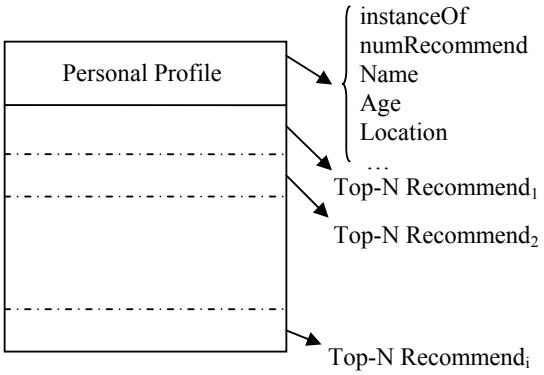

Fig. 6 Customer profile

B. Customer-Product Rating Matrix

Most of the model-based recommender systems use the product-product matrix which represents the relationships between products. Product-customer matrix is another choice for these recommender systems where the preferences of each customer to each product are determined by this matrix. Computation complexity of these matrixes linearly grows with the number of customers and products. So, computation of these matrixes requires a lot of time and space.

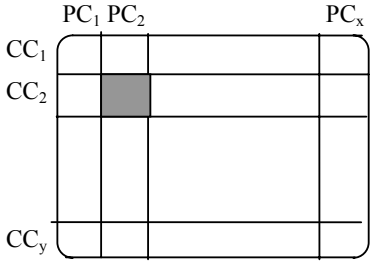

Fig. 7 Customer-Product Rating Matrix

Our proposed solution to this scalability lack is utilizing the classification of products and customers. The proposed Customer-Product Rating (CPR) matrix involves the preferences of customers in class *i* to products in class *j* and will be shown as CPR [i, j]. The amount of CPR [i, j] is between zero and one that is updated dynamically at each purchasing process. When the customer enters the electronic market, it has been supposed that he/she belongs to class *a* (CC[a]) and has purchased these products:

Buy-CID-Date= $\{(i_1, n_1), (i_2, n_2), (i_3, n_3)\ldots (i_x, n_x)\}$.

Above formation shows that one purchase has occurred by a customer with *CID* code at *Date.*, *t* indicates the total number of purchasing items which is obtained by calculating $n_1+n_2+ \ldots+n_x$. In CPR matrix, row *a* contains these values with respect to purchase of current customer where these values must be refined as below:

$$CPR[a][] = [(\alpha 1 + n1)/(100 + t), (\alpha 2 + n2)/(100 + t), \ldots..]$$

C. Customer Class Discovery

A formal definition of classification will not be attempted; for our purposes it is sufficient to think of classification as describing the process by which a classificatory system is constructed. The word *classification* is also used to describe the result of such a process. Although indexing is often thought of as classification we specifically exclude this meaning. A further distinction to be made is between classification and *diagnosis*. Every language is very ambiguous on this point:

- How would you classify (identify) this?
- How are these best classified (grouped)?

The first example refers to diagnosis whereas the second talks about proper classification [23]. In this section we study how to diagnose the class of one object.

VSM[1] has been selected as the best approach to diagnose the class of one object [22, 24]. It is standard technique in information retrieval. The VSM allows decisions to be made about to which class each object belongs. It follows these steps:

- Users select attributes of the current object from among of the all attributes and assign values to them.
- Equivalents of each attribute will be determined using WordNet ontology and one set is created for each attribute.
- Classes of catalog that have the same attributes as current object is determined.
- The cosine similarity between selected classes and current object is calculated.
- User selects the best class from among of output classes.

The next subsection describes the proposed recommendation algorithm.

D. Making Recommendation

One of the most remarkable characteristics of a successful algorithm is the range of products that is recommended by that. Hence, in the proposed algorithm, we divide Top-N-Recommendation between overall product classes. This algorithm recommends some of the products in each class of product classes based on the rating amounts in CPR matrix. Suppose that row *a* of CPR matrix contains these values,

$$CPR[a][] = [\alpha 1, \alpha 2, \alpha 3, \ldots., \alpha x]$$

[1] Vector Space Model

The algorithm recommends $\alpha 1$*100 percent of Top-N-Recommendation from PC_1 class. This means that α_1*N percent of recommendations in Top-N-Recommendation belong to products in class PC_1. Similarly, $\alpha 2$*100 percent of Top-N-Recommendation belong to products in PC_2 class and so on.

The customer sequence in the proposed recommendation system is shown in Fig. 8. In the next section, our proposed method is investigated doing some evaluation tests.

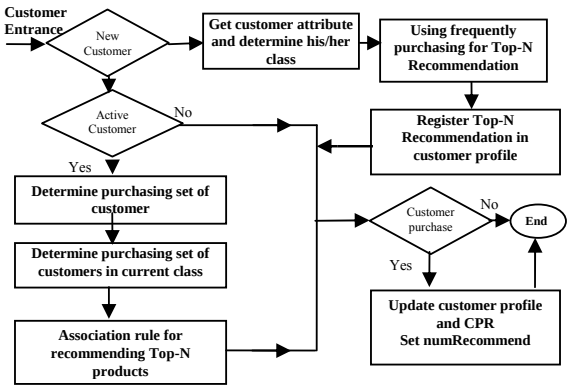

Fig. 8 Customer sequence in e-market

V. Evaluation

A. Dataset

We used data from our MovieLens recommender system, MovieLens is a web-based research recommender system that debuted in fall 2003 [19]. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies. The site now has over 6000 users who have expressed opinions on 4000 movies. We randomly select enough users to obtain 200,000 ratings from the database. We divided the database into 80% training set and 20% test set. We performed our procedure on the training set and compared its results with test set.

B. Evaluation Metric

To evaluate top-N recommendation, a widely used metric in information retrieval (IR) community, namely *precision* [21] has been utilized. But, we have slightly modified the definition of precision as our experiment in the sense that we have a fixed number of recommended items. In the first step, the algorithm works on the training portion of data where *top-N set* will be generated as the set of recommendations (for each customer). Our main goal is to look into the test set and match the newly recommended products with previously generated top-N set. Products that appear in both sets are members of a special set, which is called the *hit set*. We now define precision metrics in our context as bellow:

- *Precision.* Which is defined as the ratio of hit set size to the top-N set size, i.e., *precision= size of hit set/ size of top-N set.*
- *Recall.* We define recall as the ratio of hit set size to the test set size, i.e., *recall=size of hit set / size of test set.*

These two measures are, often conflicting in nature. For instance, increasing the number N tends to increase recall but decrease precision. The fact that both are critical for the quality judgment leads us to use a combination of them. In particular, we use the standard *F1 Metric* that gives equal weight to both of them and is computed as

F1= 2*Recall*Precision/ (Recall+ Precision)

We compute F1 for each individual customer and calculate the average value as our metric.

Coverage measures the percentage of the universe of items that the recommendation system is capable of recommending. For the prediction task, it is calculated as the percentage of unrated items, a rating for which can be predicted by the system. An alternative is to calculate coverage as the percentage of items of interest to a user rather than considering the complete universe of items.

C. Experimental Results

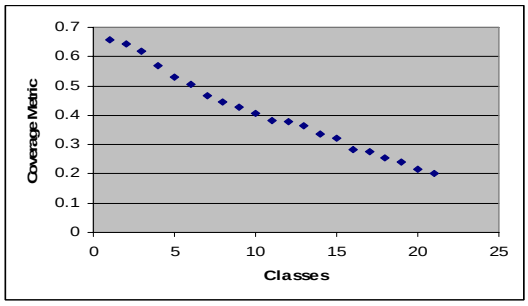

Fig. 9 Coverage metric for all customers

In this section we present and discuss the evaluation results for this recommender algorithm. The algorithm utilizes the data set described above as input and the produced recommendations were evaluated using the useful recommendations identified by two types of customers. First, we evaluate our algorithm for recommending to new customers. Afterwards, it is evaluated for customers that have earlier purchase.

Considering the coverage metric, the result of evaluating our algorithm for two types of customers has been shown in Fig. 9. We calculate coverage metric for the number of the classes of this dataset. Coverage metric change between 0.7 to 0.2 and means our algorithm be able to recommend 20 to 70 percent of the all products from 21 product classes.

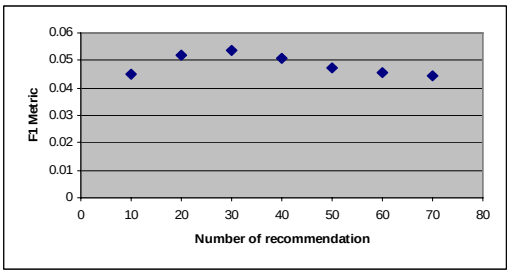

Fig. 10 F1 Metric for new customers

Considering F1 metric, the results of the custom implementation of our algorithm are presented in Fig.10. These results have been acquire when the customer is new and had not have any purchases in the past. As you can see, F1 metric is optimal when the number of recommendation is around 30.

This algorithm can recommend better product to customer when it knows the previous actions of the customer. It will be determined purchasing set of the customer and customers in his/her class. The algorithm uses association rules for specifying recommended products. We apply Apriori algorithm for mining association rules and its result will be recommend to customer. As you can see in Fig.11 recommending products to earlier customers have better results, considering F1 metric. F1 metric will increase as the number of recommendations increase.

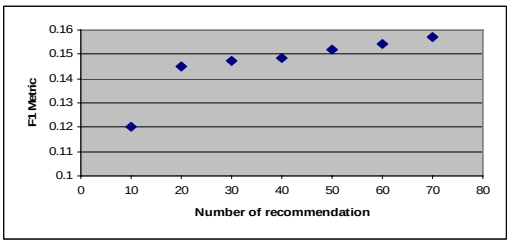

Fig. 11 F1 Metric for Earlier customers

VI. Conclusion

Recommender systems benefit customers by enabling them to find products they like. Conversely, they help the business by generating more sales. These systems are rapidly becoming a crucial tool in E-commerce on the web. New technology are needed that can dramatically improve the scalability of recommender systems.

In this paper, we introduce a procedure for recommending products to customers. We model products and users information using OWL ontology language. Using this approach, sellers recommend semantic object to customers. Our algorithm do not have 'New Item' problem since we classify the products and the users based on their property. The experimental results have shown superior performance in terms of coverage and precision.

References

[1] B.M. Sarwar, G. Karypis, J.A. Konstan, J.T. Riedl, "Item-based Collaborative Filtering Recommendation Algorithms", In Proc of WWW 10, 10th International World Wide Web Conference, 2001, pp. 285-295.

[2] M. Deshpande and G. Karypis. "Item-based top-n recommendation algorithms". ACM Trans. Inf. Syst., 2004, pp.143-177.

[3] J.K. Kim, Y.H. Cho, W.J. Kim, J.R. Kim, J.H. Suh, "A personalized recommendation procedure for internet shopping support", Electronic Commerce Research and Applications", 2002, pp. 301-313.

[4] Y.H. Cho, J.K. Kim, and S.H. Kim, "A personalized recommendation system based on web usage mining and decision tree induction", *Expert Syst. With Applications*, Vol. 23, 2002, pp.329-342.

[5] Y. Zhang and J. Jiao, "An Associative Classification-based Recommendation System for Personalization in B2C E-Commerce Applications", Expert Systems with Applications, Vol. 33, No. 2, 2007, Forthcoming.

[6] C.H. Lee, Y.H. Kim, P.K. Rhee, "Web personalization expert with combining collaborative filtering and association rule mining technique", Expert Systems With Applications, 2001 – Elsevier.

[7] M. Balabanovic and Y. Shoham, "*Fab: Content-Based, Collaborative Recommendation*", Communications of the ACM, vol. 40, 1997, pp. 66-72.

[8] B. Mobasher, "Web Usage Mining and Personalization", in Practical Handbook of Internet Computing. M. P. Singh(eds), CRC Press, 2004.

[9] J. B. Schafer, J. Konstan, and J. Riedl. "Recommender systems in e-commerce". In Proceedings of the 1st ACM Conference on Electronic Commerce, 1999, pages 158-166.

[10] Y. Benjamin, K. Robin, "Personalization of information access for electronic catalogs on the web", Electronic Commerce Research and Applications, 2003.

[11] B. Mobasher, R. Cooley, and J. Srivastava. "*Automatic personalization based on Web usage mining*". Communications of the ACM, 2000, 142-151.

[12] B. Mobasher, S. Anand: "Intelligent Techniques for Web Personalization", IJCAI 2003 Workshop, ITWP 2003, Acapulco, Mexico, August 11, 2003, Revised Selected Papers Springer 2005.

[13] G. Antoniou and F. van Harmelen. "Web Ontology Language: OWL". In S. Staab and R. Studer (Eds), Handbook on Ontologies in Information Systems, Springer 2003.

[14] B. M. Sarwar, G. Karypis, J. A. Konstan, and John Riedl, "Analysis of recommendation algorithms for e-commerce", in Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00), Minneapolis, MN, USA, (October 2000), pp. 158-167.

[15] B. Sarwar, G. Karypis, J. Konstan, AND J. Riedl, "Item-based collaborative filtering recommendation algorithms". In WWW10, 2001.

[16] Y. Yang, B. Padmanabhan, "Evaluation of online personalization systems: a survey of evaluation schema and a knowledge-based approach", Journal of electronic commerce, 2005.

[17] G. Karypis, "Experimental evaluation of item-based top-n recommendation algorithms". In proceeding of the ACM conference on information and knowledge management. ACM, New York, 2001.

[18] R. Agrawal, T. Imielinski, And A. Swami, "Mining association rules between sets of items in large databases". In proceeding of 1993 ACM-SIGMOD International conference on Management of Data (Washington, D.C). ACM, New York, 1993.

[19] MovieLens 2003. Available at http://www.groupLens.org/data

[20] Resnick, P., Varian, H. R. (1997). "Recommender Systems". Special issue of Communications of the ACM. 40(3).

[21] G. Kowalski, "Information Retrieval Systems: Theory and implementation". Kluwer Academic publisher, 1997.

[22] C. Aasheim and Gary J. Koehler, "Scanning World Wide Web Documents with the Vector Space Model," forthcoming Decision Support Systems.

[23] "Three Automatic Classification", Available at www.dcs.gla.ac.uk/Keith/Chapter.3/Ch.3.html

[24] G. Salton, A. Wong, and CS Yang , "A Vector Space Model for Automatic Indexing," Communications of the ACM, vol. 18, nr. 11, 1975, pp 613–620.

[25] M.K. Smith, C. Welty and D.L. McGuinness. "OWL Web Ontology Language Guide", Available at http://www.w3.org/TR/owl-guide/

[26] H. khosravi Farsani, M.A Nematbakhsh, "Modeling Electronic Product Catalog using OWL Ontology Language", in Proceedinf of IADIS Conference, 2006.