Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)

Exploiting Semantic Descriptions of Products and User Profiles for Recommender Systems

Pingfeng Liu
School of Economics
Wuhan University of Technology
P.R. China
lpf@mail.whut.edu.cn

Guihua Nie
School of Economics
Wuhan University of Technology
P.R. China
niegh@mail.whut.edu.cn

Donglin Chen
School of Economics
Wuhan University of Technology
P.R. China
chendl@mail.whut.edu.cn

***Abstract*-To enable semantics based recommender systems, products and user profiles need to be represented in knowledge uniformly where ontology can be exploited. Product ontology describes the attributes of the product such as appearance, structure, behavior and function, and has a property "service" which describes the services related to the product supplied by the products provider. So service ontology need to be constructed due to its great influences on users when they browse and purchase products. User profile is modeled as a set of triple <*Goal*, *Constraint*, *Preference*> where *Goal* is the product a user searches for, *Constraint* indicates the conditions a user prescribes that must be satisfied by the attributes of the goals and *Preference* indicates users' preferences in specific dimensions of the attributes of the goals. The con*straint* and *preference* in product attributes are obtained through mining user's past browsing behaviors and transaction records. The mining algorithm is given in this paper. The method of implicit rating and weight evaluation of product attributes are also explored in this paper. A hybrid approach combining semantic similarity with collaborative filtering is proposed to generate the recommendation lists for users where the semantic similarity algorithm is adopted to get the nearest neighbors of the active user. The experiment results are presented which demonstrate that our approach is feasible.**

I. Introduction

Recommender systems help online businesses enhance quality of service and increase sales while helping customers ease their information overload through automatically applying personalized recommendations for products to customers during a live interaction. Recommender systems are mostly designed based on content-based filtering or collaborative filtering. The content-based filtering approach makes recommendation by analyzing the information content and matching keywords or classifications [1]. This technique suffers from two weaknesses, namely content limitation and over-specialization [2]. The collaborative filtering (CF) approach overcomes those two shortcomings for it does not use the actual content of the items but reference other users' access behaviors for recommendation. By adopting nearest-neighbor algorithm, CF recommender systems evaluate the similarity between users based on their ratings of products, and make recommendation considering the items visited by nearest neighbors of the user. However CF technique suffers from the problems of scalability and sparsity [3]. We noticed that the recommended lists generated by the above techniques might have low cover rates or call rates, and sometimes seemed misleading. Most often keywords or classification or product ratings are far from representing user requirements and interests. Semantics need to be incorporated into recommender systems.

In this paper, we propose a hybrid approach combining semantic similarity with collaborative filtering to generate recommendation lists for users. The next section provides an overview of the related works. Section Ⅲ discusses how to represent product and service with ontology. Section Ⅳ describes the description model of user profile, explores the method of implicit rating and weight evaluation of product attributes, and proposes the algorithm of mining user profiles from user's browsing and purchasing behavior. Section Ⅴ proposes the hybrid recommendation algorithm based on semantic similarity and collaborative filtering. Section Ⅵ presents the experiment results and section Ⅶ concludes this paper.

II. Related Works

Semantic web suggests the annotation of web resources with machine-processable metadata, which can provide tools to analyze meaning and semantic relations between documents and their parts. Ontology allows the explicit specification of a domain discourse, which permits to access to and reason about an agent knowledge, incorporate semantics into the data, and promote its exchange in an explicit understandable form. Semantic web and ontology are therefore fully geared as a valuable framework for distinct applications, namely business applications like E-Commerce [4]. RDF and OWL are semantic web standards that provide a framework for asset management, enterprise integration and sharing and reuse of data on the web. W3C issued latest RDF and OWL recommendations on 10 February 2004 [17].

Product information modeling is mainly researched in such fields as virtual manufacturing and e-commerce. Xiangjun Fu, et al gave the analysis of the function which ontology takes in product knowledge expression stack [5]. They also explored the methodology of semantic representing of product data in

This research was supported by the National Natural Science Foundation of China under Grant 70572079.

1-4244-0705-2/07/$20.00 ©2007 IEEE

XML[6]. EXPRESS is a modeling language for STEP which is oriented towards product design and manufacturing. The ultimate goal is for STEP to cover the entire life cycle, from conceptual design to final disposal, for all kinds of products [18]. While the standards cater for the needs of manufacturing industry, they seem too complicated for customers who need not know the much information irrelevant to their interests when they make their decisions on purchasing. Hyunja Lee, et al took an Extended Entity Relationship approach to denote the fundamental set of modeling constructs, and present corresponding description logic representation for each construct. They model e-catalog with entities and relationships which are classified into four classes including inclusion, attribution, synonym and antonym [7]. However their model mainly focused on constructs instead of attributes which exert great influences on user's decision making when they browse and order products online.

User profile modeling plays a key role in recommender systems. We proposed to model user requirements with a triple <*Goal*, *Constraint*, *Preference*> and the user requirements ontology is constructed accordingly adopting OWL. We also discussed how to retrieve user requirements explicitly through user interaction with the aid of external linguistic resources and ontology base, make recommendations based on the semantic matching between the user requirements ontology and product ontology [8]. Similarly the conversational recommendation discovers user preferences from minimal information input of users step by step [9]. Generally users are not willing to interrupt their normal pattern of browsing to enter explicit ratings without benefits. To solve the problem of ratings sparsity implicit ratings need be mined from users' browsing behavior and transaction records. Stefan Holland, et al presented a novel approach for mining preferences from user log data based on the concept of strict partial order preferences [10]. Seonho Kim, etc employ a user tracking system and a user modeling technique to capture and store users' implicit ratings in their personalized digital library recommender system based on collaborative filtering. [11]. It is discovered through experiment and statistical analysis that time and scrolling events are effective implicit interest while mouse movement and mouse clicks by themselves are not [12]. However those implicit interests detecting methods mainly focus on product items other than the attributes of them, which can not refine user's preferences to the level of various attributes of the product.

Recent research work on recommender systems exhibits efforts in incorporating semantics into recommendation. In [13] a research project on resolving semantic differences for multi-agent systems (MAS) in electronic commerce is described. In [14] Cai-Nicolas Ziegler targets the successful deployment and integration of recommender system facilities for Semantic Web applications. His approach mainly builds upon the notions of taxonomy-driven interest profile assembly and trust networks.

Our work focuses on the hybrid approach combining semantic similarity with collaborative filtering to improve the effect of recommender systems. Product modeling is based on the analysis of customers' information needs while taking service into consideration. User profiles are modeled as a set of triples <*Goal*, *Constraint*, *Preference*> of which the attribute values and weights need to be mined from customers' browsing and purchasing history before they are exploited in the recommendation generating process.

III. PRODUCT MODELING

The widely used information models for product knowledge representation include the STEP standards and Function/Behavior/ Structure model. But they are mainly oriented towards product design and manufacturing. So it is essential to identify the knowledge that the customers intend to acquire by accessing a specific product concept or information element before they make their final decisions on purchasing in an e-marketplace. Thus, domain experts need to organize product information elements in a way that closely links them with their intended use. This early identification of information usage enables the delivery of exact and accurate information to users.

When a customer browses a product, basically s/he wants to understand such features concerning the product as the manufacturer, appearance, content or structure, behavior, function, performance, quality, price, payment, logistics, maintenance, and stipulations in contract. Among those information appearance, content or structure, behavior, function, performance, quality and price are modeled in product ontology while payment, logistics, maintenance, and stipulations in contract are modeled in service ontology. There are two kinds of semantic relationships in product ontology. One is attribute and another is inclusion which includes "part-of" and "is-a". Fig. 1 illustrates a simple model for the motor vehicle in UML which is then translated into the OWL syntax. In the figure the line with triangle head denotes "is-a" relationship, line with diamond head denotes "part-of" relationship while line with arrow head denotes attribute relationship. Every product has a service attribute which clarifies all the information on the services provided by the product manufacturer and vendor. We mean the attributes of both product and the related service when we mention the attributes of product in the following of this paper.

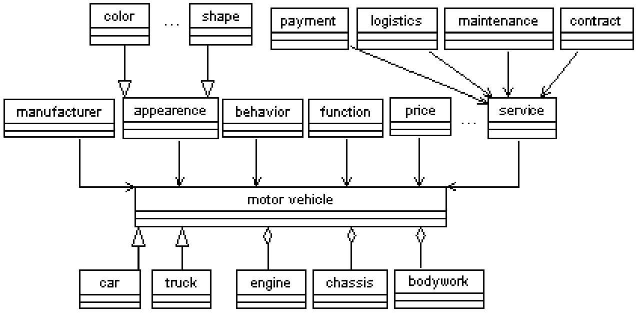

Fig. 1. A simple motor vehicle model in UML

IV. User profile modeling

In recommender systems a user is characterized by his interests. Our early work [8] modeled user requirements as a triple <*Goal*, *Constraint*, *Preference*> where *Goal* is what a user searches for, *Constraint* indicates the conditions a user prescribes that must be satisfied by the goals and *Preference* indicates users' preferences in specific dimensions of the goals. *Constraint* and *Preference* are measured on the attributes of products and the services related to the product, as well as user's resources. We also discussed how to retrieve user interests through user interaction such as input and choices explicitly. Since users are generally reluctant to rate products explicitly without benefits for doing so, rating sparsity exists and remains as a main obstacle to the performance of collaborative filtering. To understand user's interests implicitly will help alleviate the problem. Usually data mining technologies are employed to discern frequent items, associate items, clustered items, clustered users and user's browsing patterns. Then the recommendation is made by similarity between items or users, or by association rules. Different from this approach, we employ the modeling ontology, i.e. the triple <*Goal*, *Constraint*, *Preference*>, to express user's profile. Data mining technologies are used to discover user's constraints and preferences in the product attributes from web application log after it is preprocessed as is stated in [15].

A. Information Model

The information model is defined as following:

- Product domain is denoted by set P={P_m} where m=1,2,…,M and P_m is modeled as an ontology as described in section Ⅲ. According to the "is-a" semantic relationship between the products, the product class tree can be constructed from the product set.
- Attribute domain is denoted by set A={A_l} where l=1,…,L. Each attribute has a property "value" which may be of numerical value or interval, or text value, or null if the attribute consists of other sub attributes. According to the "is-a" semantic relationship between the attributes, the attribute class tree can be constructed from the attribute set.
- Product attributes are described in the product ontology. For the convenience of discussion, we denote the product attributes by the set A_{Pm}={A_1,…,$A_{L'}$}, where $A_{Pm} \subseteq A$ and $L' \leq L$.
- User domain is denoted by set U={u_1,…,u_N}.
- User rating is a matrix R={r_{ij}} where r_{ij} is user u_i's rating on product p_j. i=1,…,N and j=1,…,M.
- User profile is denoted by set UP={UP_1,…,UP_k}, where UP_k =<*Goal*, *Constraint*, *Preference* > which is called profile item. Here the *Goal* is a product class P_m from product domain while *Constraint* and *Preference* are set of triple <A_l,W_l,V_l> where A_l is an attribute class from attribute set A_{Pm} in correspondence to the Goal, W_l is the weight of user's interests in the attribute A_l and V_l is the normalized value of the attribute A_l that must be satisfied or a user prefers, i.e.

Constraint={<A_1,W_1,V_1>, …, <A_C,W_C,V_C>}

and *Preference*={<A_1,W_1,V_1>, …, <A_R,W_R,V_R>}.

B. Preference Measurement

Hypothesis: a user have browsed and analyzed all the information s/he cares about before s/he make decisions. In other words, the user's behavior is based on the rational analysis.

Based on this hypothesis, we deduced that a user will only purchase an item among the browsed ones. That is, the ranges of the values of product attributes are intervals or sets containing the browsed attribute values which constitute the constraints on the attributes. The purchased items reflect the user's preferences.

User preferences are measured by a set of <A_l,W_l,V_l>, where A_l is an attribute of the purchased items falling into the same product class as the *Goal*. The value of A_l need to be rectified to show user's preference towards the attribute and then is called preference value. Suppose $p_{purchased}$ is the set of the purchased items falling into the same product class as the *Goal*. Considering the following two different conditions:

(1) The value of the attribute A_l is numerical. Then the preference value of A_l is calculated by the following equation:

$$v_l^{'} = \begin{cases} \dfrac{v_l - \dfrac{v_{\max}^l + v_{\min}^l}{2}}{\dfrac{v_{\max}^l - v_{\min}^l}{2}} & if \quad v_{\max}^l \neq v_{\min}^l \\ 0 & if \quad v_{\max}^l = v_{\min}^l \end{cases} \tag{1}$$

Here v_l is the arithmetic mean of the values of attribute A_l of the purchased items in $p_{purchased}$ while [$v_{\min}^l, v_{\max}^l$] is the value range obtained from the corresponding attribute ontology. $v_l^{'}$ =1 means user prefer the maximum value of the attribute, $v_l^{'}$=-1 means user prefer the minimum value of the attribute while $v_l^{'}$=0 means user prefer the mean value of the attribute. If the attribute A_l does not exist in the *Preference* element of corresponding profile item, then a new triple <A_l,W_l,$v_l^{'}$> is added to the *Preference* element. Otherwise the preference value of attribute A_l is calculated by the following equation:

$$v_l^{'} = \alpha v_{lNew} + (1-\alpha) v_{lOld} \tag{2}$$

Here v_{lNew} is the preference value calculated using (1) according to user's most recent browsing and purchasing behavior, v_{lOld} is the existed preference value of attribute A_l, and α is called decay factor. The larger the value of α, the smaller the influence of user's past preference.

(2) The value of the attribute A_l is not numerical. Then the preference value of the attribute A_l is a set V_l={v_l} where v_l is the value of attribute A_l of the purchased items in $p_{purchased}$, l=1,…,L'' and L'' is the count number of distinct values of the attribute A_l of the purchased items in $p_{purchased}$. If the attribute A_l does not exist in the *Preference* element of the corresponding profile item, then a new triple <A_l,W_l,V_l> is added to the *Preference* element. Otherwise each member of V_l is appended to the corresponding preference value set of attribute A_l in the profile item if the preference value set does not include it. The non-numerical value is a set with fixed

volume which is set to 4 in our experiment. The first-in-first-out policy is adopted to make room for the new comers in case there is no vacancy for them.

Attribute weight is rectified in the same way as numerical attribute value:

$$w_l^{'} = \alpha w_{lNew} + (1-\alpha) w_{lOld} \tag{3}$$

C. Attribute Weight Calculation

Attribute weight shows the degree of user's interests in the attributes of product items when s/he browses web pages. When a user prepares to buy a product, s/he will browse the information relating to it first and compare different instances according to his or her preferences towards various attributes before s/he pays for it. So the attribute weight is largely dependent on the time a user spends on browsing the attribute. The time a user spends on browsing products and their attributes can be acquired through analyzing the web application log. If a web page contains more than one attribute, the time spent on each attribute is the time spent on the page divided by the number of attributes. If the same attribute is browsed more than once, the time spent on browsing the attribute each time is summed.

Suppose a user wants to buy a product p_i, set $A=\{A_i^l\}$ denotes all the attributes hold by the product p_i and $l=1,\ldots,L'$. The number of web pages relating to product p_i browsed by the user amounts to S, the time spent on browsing the sth page is t_s, $f_s^{il}=1$ denotes the sth page browsed by the user contains the attribute A_i^l while $f_s^{il}=0$ denotes the sth page browsed by the user does not. t_i^l is the time user spends on browsing the attribute A_i^l. W_i^l is the weight of attribute A_i^l. r_i^l is user's implicit rating on attribute A_i^l while r_i is user's implicit rating on product p_i. The implicit ratings on product and its attributes and the weight of product attributes can be calculated by equation (4), (5), (6) and (7).

$$t_i^l = \sum_{s=1}^{S} \frac{t_s \times f_s^{il}}{\sum_{l=1}^{L'} f_s^{il}} \tag{4}$$

$$w_i^l = \frac{t_i^l}{\sum_{l=1}^{L'} t_i^l} \tag{5}$$

$$r_i^l = \frac{t_i^l}{\sum_{s=1}^{S} t_s} \tag{6}$$

$$r_i = \sum_{l=1}^{L'} w_i^l \times r_i^l \tag{7}$$

D. User Profile Mining Process

User profile is obtained by analyzing and mining user browsing and purchasing behaviors. A user browses web sites for the information of target object and balances the different choices before s/he makes his or her final decision. It is noticed that user may change his or her interests with the elapse of time. So user profile should be maintained accordingly when the change is detected.

The user profile is generated through mining user's browsing behavior and transaction records. The algorithm is as following:

Input: user browsing path and purchased items, old user profile *UP*.

Process:

Let $p^{'}$ be a set of which each member P_i is a set of product items belonging to the same product class; let $A^{'}$ be a set of which each member A_i is a set of attributes corresponding to P_i; let $W^{'}$ be a set of which each member W_i is a set of attribute weights corresponding to A_i; let $V^{'}$ be a set of which each member V_i is a set of attribute values corresponding to A_i.

```
MineUserProfile
{
  For each page browsed by the user
  {
    Create or modify $p^{'}$, $A^{'}$, $V^{'}$, $W^{'}$ corresponding to product item $p_i^j$
      extracted from the page where equation (4) and (5) are used to
      calculate the attribute weights
  }
  For each product class $P_i$ in $p^{'}${
    SetConstraint($UP_k, p_i, A_i, W_i, V_i$)
    Set Preference($UP_k, p_i, A_i, W_i, V_i$)
  }
}
:SetConstraint($UP_k, p_i, A_i, W_i, V_i$)
{
  If attribute $A_i^l$ does not exist in the constraint of $UP_k$, add a new triple
      $< A_i^l, w_i^l, v_i^l >$ to it
  If the value of $A_i^l$ is numerical, adjust interval [$v_{\min}^l, v_{\max}^l$] which is the
      value of attribute $A_i^{l'}$ in Constraint of $UP_k$ by modifying $v_{\min}^l$ or
      $v_{\max}^l$ if the corresponding value in the value set of attribute $A_i^l$
      falls outside it
  Else if any member in value set of $A_i^l$ does not exist in the value set of
      corresponding attribute in the Constraint of $UP_k$, append it to the
      value set
  Calculate the attribute weight in the Constraint of $UP_k$ using (3)
}
:SetPreference($UP_k, p_i, A_i, W_i, V_i$)
{
  If attribute $A_i^l$ of purchased items does not exist in the Preference of
      $UP_k$, add a new triple $< A_i^l, w_i^l, v_i^l >$ to it
  Calculate the preference value in the preference of $UP_k$ using the
      methods as stated in the above subsection B of this section
  Calculate the attribute weight in the Preference of $UP_k$ using (3)
}
```

Output: new user profile *UP'*.

V. Recommending Process

A. Neighborhood Formation

Given two user profile items: UP_i=<G_i, Co_i, Pr_i>and UP_j=<G_j, Co_j, Pr_j>, the similarity between the two profile items is calculated as following:

Step 1: calculate the semantic similarity between the two *Goals* G_i and G_j.

$$sim(G_i,G_j)=\frac{2\times depth(lso(G_i,G_j))}{len(G_i,lso(G_i,G_j))+len(G_j,lso(G_i,G_j))+2\times depth(lso(G_i,G_j))} \quad (8)$$

where *lso*(G_i, G_j) denotes the nearest common ancestor of G_i and G_j, *depth*() denotes the distance from the root of the product class tree, *len*() denotes the distance between G_i and G_j in the product class tree.

Step 2: calculate the semantic similarity between two *Preference* items Pr_i and Pr_j.

The semantic similarity between two attributes A_k and A_l is calculated by the following equation in the same way as Goals:

$$sim(A_k,A_l)=\frac{2\times depth(lso(A_k,A_l))}{len(A_k,lso(A_k,A_l))+len(A_j,lso(A_k,A_l))+2\times depth(lso(A_k,A_l))} \quad (9)$$

The similarity of weights between two attributes is calculated as:

$$sim(w_k,w_l)=1-|w_k-w_l| \quad (10)$$

The similarity of values between two attributes is calculated as:

When attribute value is numerical,

$$sim(v_k,v_l)=1-|v_k-v_l| \quad (11)$$

Otherwise attribute value is a set,

$$sim(v_k,v_l)=\frac{\#(v_k\cap v_l)}{\#(v_k\cup v_l)} \quad (12)$$

where #() denotes the cardinality of the set.

Given two triple E_k=< A_k, w_k, v_k > and E_l=< A_l, w_l, v_l >, the semantic similarity between them is calculated as:

$$sim(E_k,E_l)=sim(A_k,A_l)\times[\beta sim(w_k,w_l)+(1-\beta)sim(v_k,v_l)] \quad (13)$$

where β satisfies $0<\beta<1$. In our experiments, β=0.5

Suppose there are K attributes in Pr_i and L attributes in Pr_j. Let $sim(\Pr_k,\Pr_l)$ =0. Establish the semantic similarity matrix of the two *Preference* items Pr_i and Pr_j as:

$$SP=\begin{pmatrix} sp_{11} & sp_{12} & \cdots & sp_{1L} \\ sp_{21} & sp_{22} & \cdots & sp_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ sp_{K1} & sp_{K2} & \cdots & sp_{KL} \end{pmatrix}$$

where $sp_{kl}=sim(A_k,A_l)$, which is calculated using (9).

Get the maximum sp_{kl} from the matrix and then delete the row and column the maximum sp_{kl} is located in. Locate the two corresponding triples in Pr_i and Pr_j by the indexes in the matrix and calculate the semantic similarity $sim(E_k,E_l)$ between the two triples using (13). Let $sim(\Pr_k,\Pr_l)=sim(\Pr_k,\Pr_l)+sim(E_k,E_l)$. Repeat this process until there is no elements left in the matrix *SP*.

Finally we get the semantic similarity between Pr_i and Pr_j as:

$$sim(\Pr_i,\Pr_j)=sim(\Pr_i,\Pr_j)/\min(K,L) \quad (14)$$

Step 3: calculate the semantic similarity between UP_i and UP_j as following:

$$sim(UP_i,UP_j)=sim(G_i,G_j)\times sim(\Pr_i,\Pr_j) \quad (15)$$

Now we can calculate the semantic similarity between two users which is represented by the similarity between their profiles. Suppose user u_i has *M* profile items and user u_j has *N* profile items, the similarity between u_i and u_j is calculated in the same way as two *preference* items. Let $sim(u_i,u_j)$=0

$$SG=\begin{pmatrix} sg_{11} & sg_{12} & \cdots & sg_{1N} \\ sg_{21} & sg_{22} & \cdots & sg_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ sg_{M1} & sg_{M2} & \cdots & sg_{MN} \end{pmatrix}$$

where $sg_{mn}=sim(G_m,G_n)$, which is calculated using (8).

Get the maximum sg_{mn} from the matrix and then delete the row and column the maximum sg_{mn} is located in. Locate the two corresponding profile items of user u_i and u_j by the indexes in the matrix and calculate the semantic similarity $sim(UP_m,UP_n)$ between the two profile items using (15). Let $sim(u_i,u_j)=sim(u_i,u_j)+sim(UP_m,UP_n)$. Repeat this process until there is no elements left in the matrix *SG*.

Finally we get the semantic similarity between u_i and u_j as:

$$sim(u_i,u_j)=sim(u_i,u_j)/\min(M,N) \quad (16)$$

Neighborhood formation is followed by computing similarity $sim(u_i,u_j)$ for the active user u_i and users $u_j\in U\setminus u_i$. User u_i's neighborhood contains most similar peers for use in computing recommendation lists. There are two techniques for neighborhood selection, namely correlation-thresholding and best-M-neighbors. Correlation-thresholding picks users u_j with similarities $sim(u_i,u_j)$ above some given threshold, whereas best-M-neighbors picks the M best correlates for u_i's neighborhood. It should be noticed that correlation-thresholding implies diverse unwanted effects when sparsity prevails[16]. In our experiment best-M-neighbors is adopted.

B. Recommendation Generation

The active user u_i's personalized recommendation list is taken from its neighborhood by deriving the top-N recommendations. The methods often adopted for recommendation generation include Most-frequent-item recommendation, Association-rule-based recommendation, weighted-average-of-ratings method and deviation-from-mean method. We opt for the last method for it performs well in the situation where rating data density of different users differs much with each other. Let Neighbor(u_i) be the M best neighbors for u_i, then Candidate products are

$$P^{''}=\bigcup\{R_m\,|\,u_m\in Neighbor(u_i)\}\setminus R_i$$

where R_m is products that user u_m in Neighbor(u_i) has rated and R_i is products that the active user u_i has rated. For each product p_j in $P^{''}$ we adopt deviation-from-mean method to predict preference score of the active user u_i on the product p_j as:

$$r_{ij} = \bar{r}_i + \frac{\sum_{m=1}^{M} (r_{mj} - \bar{r}_m) sim(u_i, u_m)}{\sum_{m=1}^{M} sim(u_i, u_m)}$$

where $\bar{r}_i$ and $\bar{r}_m$ are the average scores of the active user u_i and the similar user u_m. The products with the N highest preference scores are formed to be the final recommendation list with the preference score in a descending order.

VI. Experiment Results

A. Experiment Set

We adopted the above algorithm in our simulated e-commerce recommender system for digital products. We took the product data from an online store. The system keeps track of the web application log of students' browsing and simulated purchasing behavior. The data are stored in the DB2 database system on an AS400 machine together with the transaction records, product data, product ontology, and user profile. Our recommender system includes three subsystems, namely on-line mining subsystem, recommendation subsystem and off-line learning subsystem. In recent five months more than 300 students have used this system in their simulated e-commerce transaction experiments. When each student registers into the system, s/he is given a virtual income which is of tree levels, namely high level with more than ¥5000 per month, medium level between ¥3000 and ¥5000 per month, and low level less than ¥3000 per month. The income levels change in turn when newcomers register into the system so that the students are divided into three groups with nearly the same size. The students pay the products by their virtual income.

MAE (mean absolute error) is adopted as the evaluation metric. Along with each product in the recommendation lists an evaluation is asked to be provided by the relevant student. Users' ratings on products is of five levels, that is, "strongly accept", "weakly accept", "neutral", "weakly reject", and "strongly reject", which is translated into score 5,4,3,2,1 respectively in the inner format of system data.

B. Results Evaluation

Considering the impact of decay factor α in equation (2) and (3), we conducted our experiment first by determining the sensitivity of it. From the sensitivity plot we fix the optimum value of the decay factor which is used for the rest of the experiment. The neighborhood size is set to 20. We can observe from Fig. 1 that the value of α exerts great influences on the quality of prediction which deteriorates when the value of α goes to extremes and remains relatively steady when it changes from 0.3 to 0.7 with the optimum value at 0.4. Too low value of α can't reflect the preference changes of users in time while too high value of α can't reserve the long-term interests of users.

After the optimum value of decay factor justified by the first experiment is fixed, the experiment is conducted to compare our semantics based collaborative filtering algorithm with the traditional ones. As Fig. 3 shows, the neighborhood size spans from 10 to 130. On the whole the semantics based algorithm presents better quality of prediction than the traditional ones while the best quality is exhibited when the neighborhood size is 30. It is also noticed that the quality of prediction improves fast when the neighborhood size changes from 10 to 30, but decreases a little when the neighborhood size continues to increase, which we think is caused by the negative impact of the neighbors whose semantic similarities with the target user are too low.

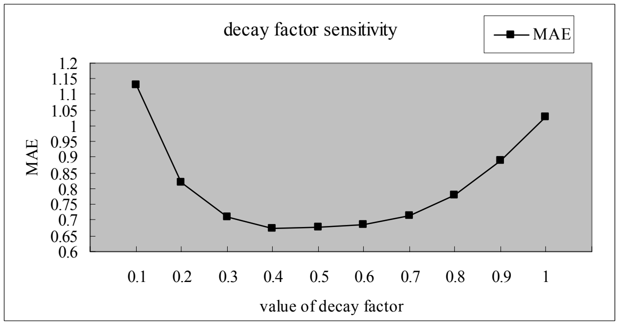

Fig. 2. The sensitivity of the decay factor

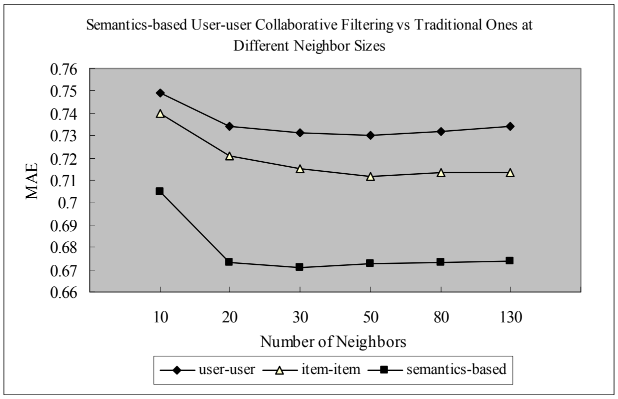

Fig. 3. The comparison of MAE on three algorithms

VII. Conclusion and Future Work

We presented a hybrid approach to generate recommendation lists for users combining semantic similarity with collaborative filtering. Instead of relying purely on the rating scores of product items, we incorporate semantics into the recommender system by modeling product and user profile as ontologies. The preference of users is refined to the level of product attributes of which the preference value and weight are taken into consideration when product-product similarity and user-user similarity are calculated. We proposed the method of implicit ratings and weights evaluation of product attributes. We also explored the mining algorithm of user profile.

In our future work we plan to improve the algorithm by clustering on different levels of product and attribute in correspondence to the product ontology to form semantic feature oriented communities so that recommendation of higher quality and performance can be generated. Another problem waiting to be solved is to improve the performance of the system while achieving a good quality when the number of concurrent users reaches a high level by incorporating the grid technology.

REFERENCES

[1] Choochart Haruechaiyasak, Mei-Ling Shyu, and Shu-Ching Chen, "A data mining framework for building a Web-page recommender system", *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration*, 8-10 Nov 2004, pp. 357–362.

[2]CYRUS SHAHABI, and YI-SHIN CHEN, "An Adaptive Recommendation System without Explicit Acquisition of User Relevance Feedback", *Distributed and Parallel Databases*, Kluwer Academic Publishers, Netherlands, 2003, pp. 173–192.

[3]Qilin Li, and Mingtian Zhou, "Research and design of an efficient collaborative filtering predication algorithm", *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies,* 27-29 Aug 2003, pp. 171–174.

[4]Silva, N., and Rocha, J., "Semantic Web complex ontology mapping", *Proceedings IEEE/WIC International Conference on Web Intelligence*, 13-17 Oct 2003, pp. 82-88.

[5]Xiangjun Fu, Shanping Li, Ming Guo and Sailong He, "Ontological Driven Product Knowledge Representation", *Proceedings of the 5'World Congress on Intelligent Control and Automation*, June 15-19. 2004, pp.2809–2813.

[6]Xiangjun Fu, Shanping Li, Ming Guo and Nizamuddin Channa, "Methodology for Semantic Representing of Product in XML", C.-H. Chi and K.-Y. Lam (Eds.): *AWCC 2004*, LNCS 3309, pp. 380–387.

[7]Hyunja Lee, Junho Shim, Dongkyu Kim, "Ontological Modeling of e-Catalogs using EER and Description Logics", *Proceedings of the 2005 International Workshop on Data Engineering Issues in E-Commerce*, pp. 125–131.

[8]Pingfeng Liu, Guihua Nie, and Dongli Chen. "Towards Semantic Description of User Requiements in Recommender Systems", *Proceedings of the Fifth Wuhan International Conference on E-Business* , May 27, 2006, Wuhan, China, pp.509-515

[9]Maria Salamó, James Reilly, Lorraine McGinty, and Barry Smith, "Knowledge Discovery from User Preferences in Conversational Recommendation", A. Jorge et al. (Eds.): *PKDD 2005*, LNAI 3721, pp. 228–239.

[10]Stefan Holland, Martin Ester, and Werner Kießling, "Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications", N. Lavra_ et al. (Eds.): *PKDD 2003*, LNAI 2838, pp. 204–216.

[11]Seonho Kim, Uma Murthy, Kapil Ahuja, Sandi Vasile, and Edward A. Fox, "Effectiveness of Implicit Rating Data on Characterizing Users in Complex Information Systems", A. Rauber et al. (Eds.): *ECDL 2005*, LNCS 3652, pp. 186–194.

[12]Mark Claypool, "Implicit Interest Indicator", *IUI'01*, January 14-17, 2001, pp. 33–40.

[13]Yun Peng, et al., "Semantic Resolution for E-Commerce", W. Truszkowski, C. Rouff, M. Hinchey (Eds.): *WRAC* 2002, LNAI 2564, pp. 355–366.

[14]Cai-Nicolas Ziegler, "Semantic Web Recommender Systems", W. Lindner et al. (Eds.): *EDBT 2004Workshops*, LNCS 3268, pp. 78–89.

[15]Cooley, R., Mobasher, B., and Srivastava, J, "Data preparation for mining world wide web browsing patterns". *Journal of Knowledge and Information Systems*. 1 (1999), pp. 5–32.

[16]CaiNicolas Ziegler, Lars SchmidtThieme, and Georg Lausen, "Exploiting Semantic Product Descriptions for Recommender Systems", *SWIR '04, ACMSIGIR Semantic Web and Information Retrieval Workshop*, July 29, 2004, Sheffield, UK

[17]W3C (2004), "World Wide Web Consortium Issues RDF and OWL Recommendations", available at: http://www.w3.org/2004/01/sws-pressrelease.html.en.

[18]Step Tools Inc. (2004), "STEP Application Protocols", http://www.steptools.com/library/standard/step_2.html