A Personalized Product Recommender System based on Semantic Similarity and TOPSIS Method

Ziming Zeng
School of Information Management, Wuhan University
Wuhan 430072,China
zmzeng1977@yahoo.com.cn
doi : 10.4156/jcit.vol6.issue7.39

Abstract

A personalized product recommendation is an effective mechanism to overcome information overload occurred when customers conduct Internet shopping. The paper presents a personalized recommender system, which integrates semantic similarity computation and TOPSIS method. First, semantic similarity is computed by constructing semantic vector-space, in order to realize the semantic content-based filtering between the product contents and customer profiles. Besides, TOPSIS method is also utilized to construct the comparison mechanism of products by calculating the utility value of each candidate product. Finally, the experiment is conducted to evaluate its recommender quality and the results show the system can give sensible recommendations and is capable of helping customers save enormous time for Internet shopping.

Keywords: *Recommender System, Semantic Similarity, TOPSIS Method*

1. Introduction

Nowadays, the advance of Internet and Web technologies has continuously boosted the prosperity of e-commerce. Through the Internet, different products and customers can now easily interact with each other, and then have their transactions within a specified time. However, the Internet infrastructure is not the only decisive factor to guarantee a successful business in the electronic market. With the continuous development of electronic commerce, it is not easy for customers to single out products and find the most suitable ones when they are often confronted with the massive product information in Internet. In the whole shopping process, customers still spend much time in visiting a flooding of retail shops on Websites, and gather valuable product information by themselves. This process is much time-consuming, even sometimes the contents of Web documents that customers browse are nothings to do with those that they need indeed. So this will inevitably influences customers' confidences and interests for shopping in Internet.

One way to overcome the above problem is to develop intelligent recommender systems to provide personalized information services [1]. A recommender system is very useful in improving decision making in complex online shopping environments and enhancing the quality of purchase decisions. In the shopping websites, the system can help customers find the most suitable products that they would like to buy by providing a list of recommended products for each given customer. To date, there are two mainly recommendation techniques to be used to recommend products for customers in recommender systems. Content-based filtering recommends products that are similar to those that the customer preferred in the past. A content-based filtering recommender system tries to understand the similarity between the target products and the products that were highly rated by the customer previously. The main disadvantages of content-based filtering are sparsity and new user problems [1, 2,3]. Collaboration filtering aims to identify customers whose interests are similar to the target customer, and recommend products preferred by them to the target customer. However, despite its success, the widespread use of collaboration filtering has exposed some problems, among which there are so-called sparsity and cold-start problems, respectively [2,4,5].

Besides these limitations, the drawbacks of both content-based and collaborative approaches mainly involve the semantic mismatching between the product items and customer profiles, as well as the lack of effective mechanism to evaluate and compare products. Therefore, it is still difficult for a customer to sort and compare products while considering all relevant attributes of product items at the same time.

In order to overcome the problems of traditional recommender techniques, the paper presents an intelligent recommender system, which integrates semantic similarity computation and TOPSIS method. First, semantic similarity can be computed by constructing semantic vector of customer profiles and product documents respectively based on the product ontology, in order to realize the semantic content-filtering between the customer profiles and product documents. Besides, TOPSIS method is also utilized to construct the multi-attribute comparison mechanism of products by calculating the utility value of each candidate product, and single out the most suitable one for customers. Therefore, the system in the paper not only considers the semantic content-filtering problems, but also utilizes the product evaluation and comparison mechanism, in order to figure out the products that best fulfill customers' needs and provide personalized shopping services for them. The paper is organized as follows. Section 2 provides the details of the personalized recommender system, with the recommendation methodology and algorithm proposed in the system. Section 3 gives some experimental results about the recommender quality in the system, and Section 4 gives an overall conclusions.

2. The personalized recommender system

2.1. System overview

The main task of the recommender system is to realize the semantic content-filtering between the customer profiles and product documents, as well as to provide the comparison and evaluation mechanism of candidate product in order to provide decision support for customers' Internet shopping. Figure 1 gives an overview of the personalized recommender model of the system.

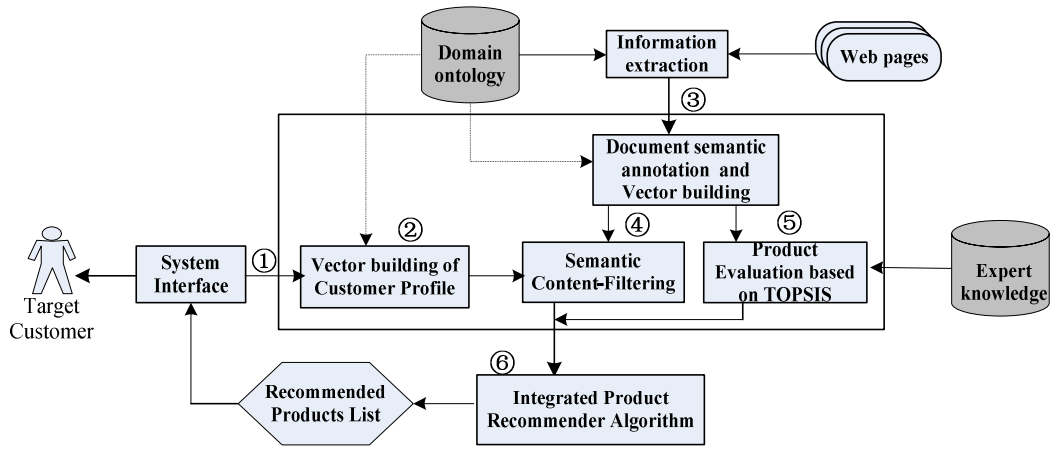

Figure 1. Overview of the recommender model of the system

The whole recommender process can be described as follows: (1) the system interface is to realize the bidirectional communication between customers and recommender system. On one hand, the system can acquire a customer's profiles by explicitly inputting information about his/her tastes, needs or preferences via the system interface. The profile information of a customer can also be elicited from his/her transactional behavior over the recent time. On the other hand, in order to collect and analyze the customer's current needs, the system interface asks him/her to express his/her qualitative needs about products. Taking a notebook for example, the system ask the customer to express qualitative features about multi-media, graphic display, network communication and interface supporting of a notebook, and give relevant weight values.(2) Once a customer profiles are acquired and extended semantically based on product ontology, the semantic vector of customer profiles can be built. (3) The web pages of candidate products are automatically extracted and the semantic vectors of product documents are built by annotating documents semantically based on ontology. (4) Once the semantic vectors of customer profiles and product documents are built respectively, semantic similarity of these

two kinds of vectors are computed, in order to realize the semantic content-based filtering between the customer profiles and product contents. (5) After extracting the qualitative feature information of annotated product documents, each candidate product is evaluated by calculating the multi-attribute utility value based on TOPSIS method. (6) The rank of each candidate product can be calculated based on the integrated product recommender algorithm, and finally the recommended products list with the highest $TOP-K$ scores are then returned to a given target customer via the system interface.

2.2. Product ontology

Ontology was originally a philosophy concept to study the essence of the existence and compositions of objectives [6]. In the artificial intelligence, ontology is a formal, shareable, and explicit specification of a shared conceptualization. Product ontology is a specialized ontology which is used to describe special domain knowledge of products. Its goal is to capture the relevant knowledge in the field, provide common understanding of the domain knowledge, identify common recognition of the concepts, and give an accurate definition for these concepts and relationships that exist between them. In the domain of product recommendation for Internet shopping, such structure can establish a good level of knowledge. In order to facilitate the search and comparison of product information provided by different e-commerce websites, a specific ontology of the particular domain must be established for the construction of various product classifications and the establishment of product information model. In terms of simple commercial websites, product ontology can be established manually or acquired semi-automatically from the contents of websites. However, in terms of large-scaled commercial websites, the manual construction of ontology is relatively complex, and the product ontology can be reused and modified based on the existing ontology, in order to adapt to the particular product domain. In the paper, the notebook ontology as the basic of the product recommendation is developed, using the Protégé editor. The simplified structure of the notebook ontology is illustrated in Figure 2.

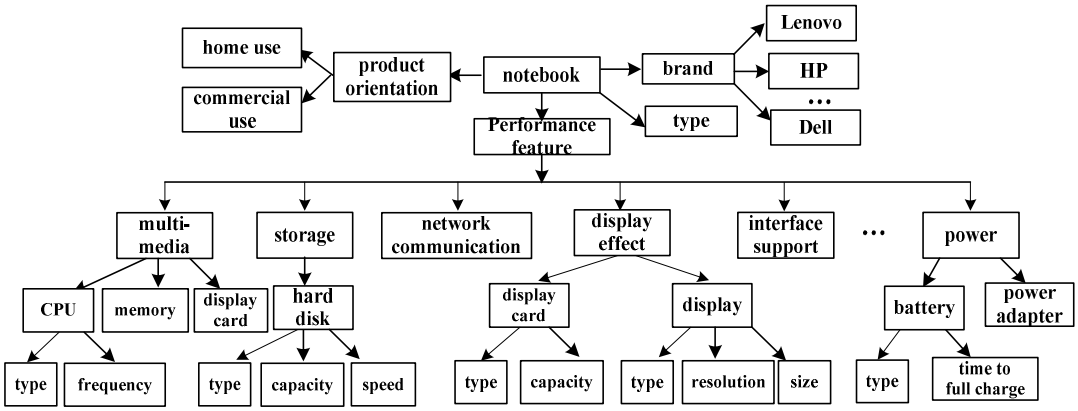

Figure 2. Overview of the recommender model of the system

2.3. Semantic vector construction of customer profiles

The customer profiles contain information about the tastes and preferences of this customer as stated before. These profiles can be obtained by analyzing the content of the items previously browsed and rated by the customer and are usually constructed using keyword analysis techniques from information retrieval [2]. For example, the profile of a customer c can be defined as a vector of weights $\Pr ofile(c) = \{W_{c1}, W_{c2}, ..., W_{ck}\}$, where each weight W_{ci} denotes the importance of keyword k_i to the customer c. Putting this method in perspective, the semantic vector construction of customer profiles can be described as followed: in the particular product recommendation such as notebooks, the advantage of the product ontology in the knowledge representation can be fully utilized to analyze and extend the initial keywords of customer profiles semantically, the aim of which is to help the system

acquire the customers' preferences and intention more accurately in the process of Internet shopping. Based on the product ontology, the semantically extended customer profiles are formed, and the semantic vector of customer profiles can be constructed correspondingly.

In the paper, the semantically extended method centers around the initial keywords, and makes the initial keywords to have synonyms extension, upper extension and lower extension respectively based on the product ontology. In the method, the synonyms extension is to acquire all the synonyms relevant to initial keywords as the new extended keywords. The upper extension is to acquire upper keywords adjacent to initial keywords as the new extended keywords. Similarly, the lower extension is to acquire lower keywords adjacent to initial keywords (including concept properties and instance values) as the new extended keywords. Therefore, when processing a customer's profiles, the system can describe and extend the initial keywords semantically based on product ontology, and acquire the customer's preference and purchase intention more accurately. For example, when a customer purchases a notebook, he can input "HP commercial use display 17 inches IEEE802.11" as initial keywords. When extracting the initial keywords, the system can utilize product ontology to decide that the customer's purchase intention is to acquire product information relevant to the ontological concept "notebook". By semantically upper extending the keywords "HP" and "commercial use" respectively, the system can deduce that notebook brand that the customer needs is HP, and the product uses is for commercial use. Similarly, by semantically lower extending the keyword "display" and upper extending the keyword "17 inches", the system can also deduce that display size that the customer needs is 17 inches. Moreover, by semantically upper extending the keyword "IEEE802.11", the system can deduce that wireless network card that the customer needs complies with the IEEE802.11 standard, and the performance feature should support wireless network communication.

After the semantic extension process of initial keywords is accomplished, the extended customer profiles can be represented as a semantic vector $Q = \{k_1, k_2, ..., k_i, ..., k_t\}$ (k_i denotes the ith keyword of customer profiles). The corresponding weight $W_i (1 \le i \le t)$ can construct the semantic vector of customer profiles, which can be defined as $WQ = \{W_1, W_2, ..., W_i, ..., W_t\}$. Each weight of the vector represents the semantic importance of relevant keyword in the customer profiles. In the construction process of semantic vector of customer profiles, the system can provide human-machine interfaces, which will help customers to further determine the accurate purchase intention in order to update the semantic vector of customer profiles in real time.

2.4. Semantic annotations and vector construction of product documents

The recommender system utilizes web crawler to choose some typical B2C websites (dangdang.com, amazon.cn, etc.). At present, a Wrapper is used to extract product attribute information from the web pages, and transfer into OWL formatted documents by taking advantage of the product ontology.

In the process of semantic document annotation, the contents of product documents or Web pages can be presented by keywords of the customer profiles. Term frequency-inverse document frequency (TF-IDF) is the best-known measure for specifying keyword weights. A keyword weight for a given product document is in direct proportion with the frequency of the keyword's appearance in the document and in inverse proportion with the number of documents that the keyword appears in. A content profile of the product document can then be represented as a vector of TF-IDF keyword weights. Formally, the semantic vector of a document can be defined as $D_k = \{C_1, C_{2,}..., C_i, ..., C_t\}$, where C_i $(1 \le i \le t)$ is a keyword of the document D_k . In the Vector-Space Model (VSM) of the documents, a keyword is often assigned a corresponding weight W_{ik} , which can be calculated by the TF-IDF scheme [7]:

$$W_{ik} = tf_{ik} \times idf_i = \frac{freq_{ik}}{\max_i freq_{ik}} \times \log(\frac{m}{n_i}) \quad (1)$$

Where $freq_{ik}$ is the number of occurrences of keyword C_i in document D_k; $\max_i freq_{ik}$ is the maximum number of occurrences of all the keywords in document D_k; m is the number of documents in the system, and n_i is the document frequency for keywords C_i in the document set D. According to the formula (1), the weights of each document in the document set D can be calculated. Therefore, in the vector space of documents, the semantic vector of document D_k can be further denoted as $D_k = \{W_{1k}, W_{2k,}..., W_{ik}, ..., W_{tk}\}$.

2.5. The Personalized product-recommendation algorithm

2.5.1. Computation of semantic similarity

Once the semantic vectors of customer profiles and product documents are constructed respectively, semantic similarity of these two kinds of vectors can be calculated, so as to realize the semantic content-based filtering between the customer profiles and product contents. The semantic similarity can be measured by using cosine similarity measure, that is:

$$sim(D_k, WQ) = \frac{D_k \bullet WQ}{|D_k| \times |WQ|} = \frac{\sum_{i=1}^{t} W_{ik} \times W_i}{\sqrt{\sum_{i=1}^{t} W_{ik}^2} \times \sqrt{\sum_{i=1}^{t} W_i}} \tag{2}$$

In the formula (2), D_k is the semantic vector of product document, WQ is the semantic vector of customer profiles, and k is the total number of keywords in the system. Obviously, the similarity value is limited in $[0,1]$. The more the similarity value is, the better the performance of semantic content-based filtering is between the customer profiles and product documents.

2.5.2. Computation of product utility value

Based on the semantic content-based filtering, the product evaluation and comparison mechanism should be established, in order to single out suitable product. The recommender system utilizes a multi-attribute decision making approach, which is derived from TOPSIS (technique for ordering preference by similarity to ideal solution) [8,9,10], to calculate the utility value of each candidate product for the customers. Its mathematical model can be described: defining $P = \{p_1, p_2, ..., p_m\}$ as the vector of the product information that the system retrieve on Internet, and $Y = \{y_1, y_2, ..., y_n\}$ as the qualitative feature vector of the products, so the utility value of the product $p_i (1 \le i \le m)$ about the attribute $y_j (1 \le j \le n)$ can be denoted as $f_{ij} = f_j(p_{ij})$, which represents the relevant performance of the product p_i in the qualitative feature i. Therefore, the decision matrix that consists of $m \times n$ f_{ij} can be defined as:

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{bmatrix} = (f_{ij})_{m \times n} \tag{3}$$

In order to facilitate mutual reference between the multi-attributes easily, the decision matrix should be normalized, which can be followed by the formula (4):

$$f_{ij}^{'} = \frac{f_{ij}}{\sqrt{\sum_{i=1}^{m}(f_{ij})^2}} \tag{4}$$

After normalizing the decision matrix, the value of $f_{ij}^{'}$ is limited in [0,1]. The utility of the product $p_i (1 \le i \le m)$ can then be calculated by TOPSIS method, which is followed by the formula (5):

$$U_i = \frac{S_i^-}{S_i^* + S_i^-} \tag{5}$$

where

$$S_i^* = \sqrt{\sum_{j=1}^{n}[\omega_j (f_{ij}^{'} - f_{j\_best}^{'})]^2} \tag{6}$$

$$S_i^- = \sqrt{\sum_{j=1}^{n}[\omega_j (f_{ij}^{'} - f_{j\_worst}^{'})]^2} \tag{7}$$

In the above equations, n is the number of product qualitative features; $f_j^{'}$ is the normalized performance value of a product in the feature dimension j; $f_{j\_best}^{'}$ and $f_{j\_worst}^{'}$ are the best and worst performance value (normalized in the same dimension, respectively; and ω_j means the customer's relative need in this feature.

TOPSIS method is based on the principle that the ideal solution should have the shortest distance to the best solution and the farthest distance to the worst one. As in the multi-attribute decision making method, the best solution represents the one with the best performance value on each of the product feature dimensions (i.e. the combination of all best ranks $f_{j\_best}^{'}$) , and the worst solution is the one with the worst performance value on each of the product feature dimensions (i.e. the combination of all worst ranks $f_{j\_worst}^{'}$) . Based on TOPSIS method, the recommender system will calculate the utility of all the candidate products, and therefore establish the comparison and evaluation mechanism of product information.

2.5.3. Integrated product recommender algorithm

Integrating semantic similarity and product utility value, the recommender score $\mathrm{Re}\,comScore_i$ of product $p_i (1 \le i \le m)$ can be computed by the formula (8):

$$\mathrm{Re}\,comScore_i = \lambda \cdot sim(D_i, WQ) + (1-\lambda)U_i \quad (0 \le \lambda \le 1) \tag{8}$$

According to the formula (8), the system will only provide the ability to have semantic content-based filtering between the product document and customer profiles when $\lambda = 1$, but can't provide the comparison mechanism of candidate product information. On the contrary, when $\lambda = 0$, the system will only provide the comparison mechanism of product information, but not considering the semantic matching between candidate product information and customer profiles. So the algorithm of integrated product recommendation can be described as follows:

Algorithm: Product_Recommendation()
Input: Customer profiles, qualitative feature needs and relative weight of commodities
Output: $TOP-k$ products list
Init(Q); Load(Q); // semantic extension of keywords of customer profiles, initiate and load semantic vector of customer profiles
Begin
for each D_i do
if log(D_i) = 0 then // document vector D_i of the product P_i is not processed
{
Load(D_i); // have semantic processing of D_i, and load it
GetSimarity(Q, D_0); // compute the semantic similarity between D_i and keywords Q in customer profiles
GetProduct_Utility(P_i); //compute the utility value of the product P_i based on TOPSIS method
GetProduct_RecomScore(P_i); // compute recommender score of the product P_i
InsertTo(ProductList); // insert P_i into ProductList in descending order
}
Output(ProductList, $top-k$); // output $top-k$ products as the recommended product to target customer
End

3. The experiment

We have developed a prototype product recommender system based on Java platform and Jakarta Lucene library. To construct the annotated data repository, the system utilizes a web crawler to search focused commodity information related to notebook from predefined B2C E-commerce websites (dangdang.com, amazon.cn,etc.) and downloads searched information to the local database. About 3000 HTML Web pages of notebook product are downloaded from relevant websites as experimental corpus, and the Wrapper is utilized to extract product information from retrieved Web pages, then the semantic annotation of documents can be realized based on product ontology.

To evaluate the quality of the recommendation product list, measures of recall and precision have been widely used in the field of recommender systems. Recall measures how many of the products in the actual customer purchase list consist of recommended products, whereas precision measures how many of the recommended products belong to the actual customer purchase list. These measures are simple to compute and intuitively appealing, however, they are in conflict, since increasing the size of the recommendation set will lead to an increase in recall, but to a decrease in precision at the same time. So a widely used combination metric called the 'F-measure' is used as the evaluation, criterion of the experiment in the paper. F-measure gives equal weight to both recall and precision, which can be computed as follow [11]:

$$F-meaure=\frac{\mathrm{Re}\,call\times \mathrm{Pr}\,ecision}{(\mathrm{Re}\,call\times \mathrm{Pr}\,ecision)/2} \tag{9}$$

In the formula (9), Recall and Precision of the recommender system can be computed respectively, according to the computing method in the literatures, so F-measure can then be computed easily. Obviously, the higher the value of F-measure is, the better the recommendation performance of the system is. However, the parameter λ in the formula (8) will influence the value of F-measure and recommendation quality of the system. In order to select the suitable value of λ, the relevant simulating experiment has been done. Based on the initial analysis of the experiment, the relation between the parameter λ and the F-measure value is illustrated in Figure 3. Seen from the Figure 3, the

F-measure value reaches the highest when $\lambda = 0.4$. Therefore, it can be inferred that the utility value computation based on TOPSIS method can give more influence on the product recommendation performance than semantic content-based filtering between the customer profiles and product documents.

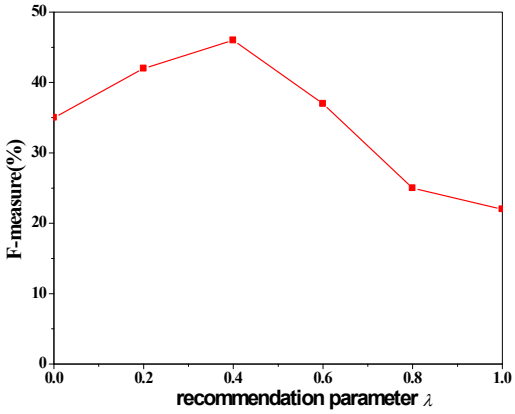

Figure 3. The relation between parameter λ and F-measure

Besides, another important issue for evaluating the recommender quality is the extent to which recommendations with higher recommender scores are accepted preferentially over recommendations with lower scores. We address this issue by comparing the distribution of scores computed from the formula (8) for accepted recommendations with the analogous distribution for offered recommendations (parameter $\lambda = 0.4$). The results are shown in Figure 4. The recommender scores for the accepted recommendations are based on 150 different kinds of notebooks accepted from 50 distinct recommendation lists. The distribution for the offered recommendations is taken from about 300 recommendations made to the customers who accepted at least one recommendation during the preliminary phase of system running.

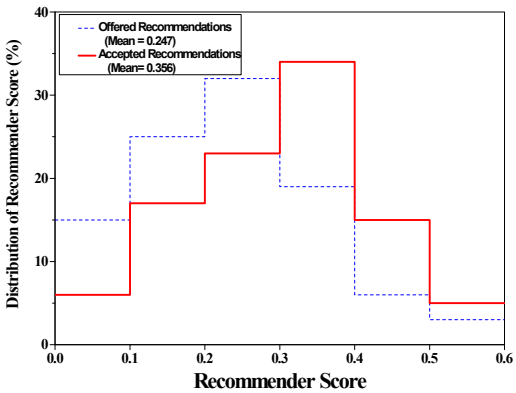

Figure 4. Distribution of scores for offered and accepted recommendations.

Figure 4 shows that the scores of the accepted recommendations are higher than the scores of a large number of offered recommendations. For example, 65% of the products placed onto the recommendations lists have scores below 0.2, but only 23% of the accepted recommendations fall in this lower span. The mean scores for the offered recommendations are 0.247, while the mean recommender scores for the accepted recommendations are 0.356. The difference between the two means is 0.109, falls well within the 85% confidence interval (0.105, 0.115) computing using t-test statistical method for the difference between means. These results illustrate that the recommender score

computed using the formula (8) is indeed a useful method of a previously unbought product's appeal to the target customer.

4. Conclusions

The paper presents an original product recommender system, which integrates the semantic similarity computation based on semantic vector-space model and product utility value computation based on TOPSIS method. The search model not only realizes semantic content-based filtering between product information and customer profiles, but also has the commodity evaluation and comparison mechanism, so as to provide the personalized product shopping services in making a successful Internet business. Furthermore, relevant experiments have been done to verify the effectiveness of product recommender algorithm in terms of F-measure criteria and by comparing the distribution of recommender scores for accepted recommendations with the distribution for only offered recommendations. The experimental results show that the algorithm presented in the paper can provide sensible recommendations and is capable of satisfying customers' shopping needs. At present, we develop a notebook recommender system based on the integrated recommender algorithm, and the system has shown some potential for e-commerce applications. Based on present research work, we will improve the recommender model and make it to provide more accurate information recommendation services for customers' shopping in the Internet.

5. Acknowledgement

The work was supported by "the Fundamental Research Funds for the Central Universities" (No: 09ZZKY096), and also supported by "the Post-70s Scholars Academic Development Program of Wuhan University".

6. References

[1] Horvath T, "A model of user preference learning for content-based recommender systems", Computing and Informatics, vol 28, no 4, pp:453-481, 2009.
[2] Adomavicius, G., and Tuzhilin.A, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", IEEE Transactions on Knowledge and Data Engineering, vol 17, no 3, pp:734-749, 2005.
[3] Burke, R, "Hybrid recommender systems: survey and experiments", User Modeling and User-Adapted Interaction, vol 12, no 4, pp:331-370, 2002.
[4] Zeng, C., Xing,C.-X., Zhou, L.-Z., & Zheng, X.-H, "Similarity measure and instance selection for collaborative filtering",International Journal of Electronic Commerce, vol 8, no 4, pp:115-129, 2004.
[5] Chen ZM, Jiang Y, Zhao Y, " A collaborative filtering recommendation algorithm based on user interest change and trust evaluation", International Journal of Digital Content Technology and its Applications, vol 4, no 4, pp: 106-113, 2010
[6] Gruber T R, "A Translation Approach to Portable Ontology Specifications", Knowledge Acquisitions, vol 5, no 2, pp: 199-220, 1993.
[7] Salton,G., "Automatic Text Processing", Addison-Wesley, New York, USA, 1989.
[8] Kaya T, " Multi-attribute evaluation of web quality in E-business using an integrated Fuzzy AHP-TOPSIS Methodology", International Journal of Computational Intelligence Systems, vol 3, no 3, pp: 301-314, 2010.
[9] Balabanovic M., Shoham Y, "Fab: content-based collaborative recommendation", Communications of the ACM, vol 40, no. 3, pp: 66-72, 1997.
[10] Jianli W, "TOPSIS method for multiple attribute decision making with incomplete weight information in linguistic setting", Journal of Convergence Information Technology, vol 5, no 10, pp: 181-187, 2010
[11] Chein-Shung, Hwang, "Genetic algorithms for feature weighting in multi-criteria recommender systems", Journal of Convergence Information Technology, vol 5, no 8, pp: 126-136, 2010