Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2008 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

July 2008

Comparision of Utility-Based Recommendation Methods

Shiu-li Huang
Ming Chuan University, slhuang@mcu.edu.tw

Follow this and additional works at: http://aisel.aisnet.org/pacis2008

COMPARISION OF UTILITY-BASED RECOMMENDATION METHODS

Huang, Shiu-Li, Ming Chuan University, Teh-Ming Road, Gwei-Shan District, 333 Taoyuan County, Taiwan, slhuang@mcu.edu.tw

Abstract

In World Wide Web environments, recommender systems are useful to reduce information overloading. A content-based recommender system recommends items according to their features. Vector Space Model (VSM) is a popular way to recommend items that are similar to those the user liked in the past. The main disadvantages of this content-based method are overspecialization and new user problems that incurred by incomplete information on user preferences. Therefore, to construct users' complete preference profiles may enhance the effectiveness of recommender systems. Some utility function elicitation methods have been developed based on Multi-Attribute Utility Theory. Whether these utility-based methods are able to outperform the traditional VSM method for recommendations is investigated in this research. This research adopts the RBFN and SMARTER methods to construct users' multi-attribute utility functions that represent their complete preferences. A laboratory experiment is conducted to compare the utility-based methods with the traditional VSM method in terms of recommendation accuracy, time expense, and user perceptions. The research results demonstrate that the VSM method is suitable to recommend items with mostly nominal attributes, and the SMARTER method is suitable to recommend items with mostly numerical attributes. The RBFN method has reliable accuracy and time expense in both recommendation contexts.

Keywords: Recommender Systems, Multi-Attribute Utility Theory, Radial Basis Function Networks, SMARTER, Vector Space Model.

1   INTRODUCTION

Recommender systems are useful to improve people's or companies' decision makings in complex environments and enhance decision ability and quality for decision makers (Resnick & Varian 1997). Generally, recommender systems use content-based or collaborative approaches to recommend interesting items to users (Adomavicius & Tuzhilim 2005).

Traditionally, content-based recommendations recommend items which are similar to those the user preferred in the past. A content-based recommender system tries to understand the commonalities between the target items and the items the user has rated highly in the past. The main disadvantages of content-based methods are overspecialization and new user problems. Collaborative recommendations identify the people whose tastes are similar to the user and recommend the items they liked to the user. The main shortcomings of this approach are rating sparsity, new user, and new item problems (Adomavicius & Tuzhilim 2005).

The drawbacks of content-based and collaborative approaches are mainly incurred by lack of ratings. If a user only rate few items, a recommender system cannot figure out the user's complete preferences by traditional methods and only can recommend items based on his/her fragmental preferences. Therefore, constructing a user's whole preference profile may enhance the effectiveness of recommender systems. However, asking users to rate all items is not feasible to build their complete preference profiles because the number of items is usually huge in a Web site or an e-marketplace. We need a feasible way to figure out a user's complete preferences.

A decision maker's preference is usually determined by many decision attributes. Multi-Attribute Utility Theory (MAUT) deals with this problem, in which, a decision maker who chooses among a number of alternatives that s/he evaluates on the basis of two or more criteria. In the field of MAUT, many methods have been developed to model a decision maker's multi-attribute utility function that can represent his/her complete preferences (Pomerol & Barba-Romero 2000). Recommender systems can be treated as a kind of decision support systems to rank alternatives (items) according to the user's multi-attribute utilities and recommends items with higher utility values to the user. MAUT motives this research to investigate whether using well-developed utility-elicitation methods to construct users' preference profiles for recommendations is able to outperform the traditional content-based recommendation approach.

2   LITERATURE REVIEW

This section briefly introduces content-based recommender systems, Multi-Attribute Utility Theory, and elicitation methods for building utility functions.

2.1   Content-Based Recommender Systems

A content-based recommender system tries to recommend items which are similar to those a given user has liked in the past (Balabanovic & Shoham 1997). Content-based recommender systems focus on how to identify the item contents, the user's interests, and the methods used to match them. Two important sub-problems exist in designing a content-based filtering system. The first is finding a content representation of items (content profiles) and the second is creating user profiles that stand for users' preferences and allow for potential items to be recommended.

Conventional content-based recommendations use vector space models (VSM) to represent user and content profiles. The contents of text-based items, *e.g.* documents or Web pages, can be represented by keywords. Term Frequency - Inverse Document Frequency (TF-IDF) is the best-known measure for

specifying keyword weights (Salton 1989). A content profile can be represented as a vector of TF-IDF keyword weights. For non-text items, the contents can be represented by attributes (Yamamoto et al. 2005). Assume that an item i has n attributes and attribute A_k ($1 \leq k \leq n$) has m_k nominal values a_1^k, a_2^k, ..., $a_{m_k}^k$. The content profile of this item can be represented as:

$$ContentProfile(i) = (a_1^1, a_2^1, ..., a_{m_1}^1, a_1^2, a_2^2, ..., a_{m_2}^2, ..., a_1^n, a_2^n, ..., a_{m_n}^n). \quad (1)$$

For a numerical attribute, its value can be an element in a content profile.

A user profile is generated by accumulating the content profile vectors of items the user has rated. Once content profiles and user profiles are encoded into vectors, the similarity between a content profile c and a user profile u is calculated as their cosine correlation:

$$Sim(u, c) = \cos\left(\vec{v}_u, \vec{v}_c\right) = \frac{\vec{v}_u \cdot \vec{v}_c}{\| \vec{v}_u \| \times \| \vec{v}_c \|} = \frac{\sum_{i=1}^{N} e_{i,u} e_{i,c}}{\sqrt{\sum_{i=1}^{N} e_{i,u}^2} \sqrt{\sum_{i=1}^{N} e_{i,c}^2}}, \quad (2)$$

where $\vec{v}_u$ is the user-profile vector, $\vec{v}_c$ is the content-profile vector, $e_{i,u}$ is the ith element in $\vec{v}_u$, $e_{i,c}$ is the ith element in $\vec{v}_c$, and N is the number of elements.

2.2 Multi-Attribute Utility Theory

Multi-Attribute Utility Theory (MAUT) is one of the major analytical tools associated with the field of decision analysis (Keeney & Raiffa 1976). A MAU function can be generally represented as:

$$MAU(u_1, ..., u_n) = \sum_{i=1}^{n} w_i \bullet u_i, \quad (3)$$

where n is the number of attribute, u_i is a single-attribute utility function over attribute i, w_i is the weight for attribute i and $\sum_{i=1}^{n} w_i = 1$ ($0 \leq w_i \leq 1$ for all i). MAUT is one of the quantitative methods that via a systematical procedure identifying and analyzing multiple variables to provide a common basis for arriving at a decision. A decision maker can calculate the utility of every alternative using the MAU function and selects the alternative with the highest utility.

A MAU function can be determined by either "holistic" or "decomposed" approaches (Schoemaker & Waid 1982). Using a holistic approach, such as multiple regression analysis (Schoemaker & Waid 1982, Laskey & Fischer 1987, Srivastava & Connolly & Beach 1995) and artificial neural networks (Malakooti & Zhou 1994, Sun & Stam & Steuer 1996; Lin & Huang & Yang 2005), a decision maker is asked to provide overall evaluations of alternatives. Using a decomposed approach, such as SMART (Edwards 1977, Edwards & Barron 1994) and AHP (Saaty 1980), a decision maker is required to compare relative importances among attributes.

This research focuses on the RBFN and SMARTER methods to design utility-based recommender systems. The RBFN (radial basis function networks) is a kind of artificial neural networks and used to solve curve-fitting (approximation) problem in a high-dimensional space. This technique has been applied to image processing, speech recognition, and time-series analysis, and firstly introduced to solve multiple criteria decision problems by Lin, Huang, and Yang (2005). Using a RBFN to model a MAU function has been demonstrated it can outperform multiple regression analysis. The SMARTER is a SMART technique, some researches have shown that SMART outperforms AHP especially when a decision problem is complex (Yap et al. 1992, Wang & Yang 1998). Moreover, AHP method must compare every two alternatives based on each attribute therefore AHP is not suitable to support decision among many alternatives.

2.2.1   SMARTER

Edwards (1977) provided Simple Multi-Attribute Rating Technique (SMART), a simple multi-criteria scoring method, to reduce the complex procedure to capture a decision maker's multi-attribute utility function. SMART was further improved to be SMART using Swing weight (SMARTS) and SMART Exploiting Ranks (SMARTER) according to different weighting methods (Edwards & Barron 1994). SMARTER uses simpler way to calculate weights and reduces a decision maker's load and time to determine the relative weights of attributes. Moreover, Edwards and Barron (1994) demonstrated that SMARTER can perform about 98% as well as SMARTS does. The main steps of SMARTER are listed as below:

Step1: Identify purpose and decision makers.

Step2: Elicit a structure or list of attributes relevant to the purpose.

Step3: Define objects of evaluation (feasible alternatives).

Step4: Formulate an objects-by-attributes matrix.

Step5: Eliminate dominated options.

Step6: Elicit single-dimension utilities. For a nominal attribute, direct rating is used to elicit its utility function. Firstly, ask a decision maker to rank all values of the attribute from the most to the least preferred. Then, the most and the least preferred values are given scores 100 and 0, respectively. Finally, ask the decision maker to rate other values of this attribute on an interval scale between 0 and 100. After that, the single attribute utility function is constructed. For a numerical attribute, bisection method (a.k.a. five-point method) is used to elicit its utility function. Firstly, the decision maker defines the two extreme attribute values that span the whole of the attribute utility range (*e.g.* 0~100). Then, the decision maker is asked to find a value that is between the two extremes and its utility is the middle of the utility range (*e.g.* 50). The decision maker further identify the "quarter values" between least preferred point to midpoint and midpoint to the most preferred point. After these steps, the single-dimension utility function of this attribute is elicited.

Step7: Rank the attributes in order of importance.

Step8: Calculate attribute weights. Rank Order Centroid (ROC) method is used to calculate weights. If $w_1 \geqq w_2 \geqq ... \geqq w_k$, and n is the number of attributes, the ROC method uses the following equation to calculate weights:

$$W_k = (1/n)\sum_{i=k}^{n}(1/i), \hspace{3cm} (4)$$

Step9: Decide. Calculate the multi-attribute utilities of the alternatives and make decision. Every alternative's utility can be computed by Formula (3). The alternative having the highest utility score will be selected.

2.3   Radial Basis Function Networks

The architecture of a RBFN is shown in Figure 1. It contains an input layer, a hidden layer, and an output layer. For an unknown function, $f(X)$: $R^n \rightarrow R$, a RBFN can approximate $f(X)$ with a set of radial basis functions. Each hidden unit, called radial basis function, is non-linear and its output for a given input X depens on the Euclidean distance between its centroid and the input. The map f is then generated by taking a weighted linear combination of these radial basis functions:

$$f(X) = \sum_{i=1}^{q} w_i \phi(\| X - C_i \|) \tag{5}$$

, where q is the number of radial basis functions, w_i and C_i is the weight and centroid of the RBF ϕ_i, respectively. A RBF typically is a Gaussian function, *i.e.*,

$$\phi(\| X - C_i \|) = \exp(-\| X - C_i \|^2 / \sigma_i^2) \tag{6}$$

, where σ_i is the width factor of the ith unit in the hidden layer.
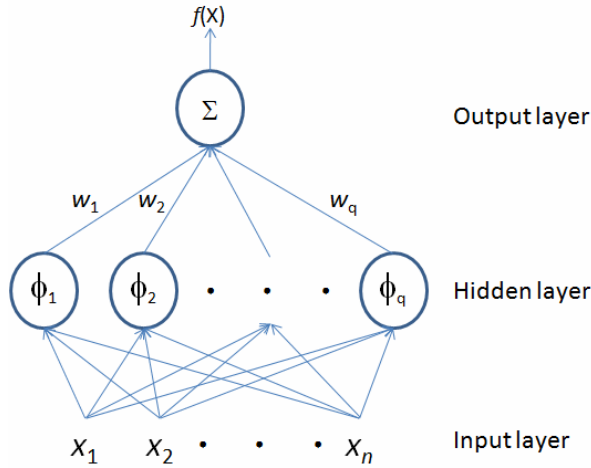

Figure 1.   *Architecture of a Radial Basis Function Network.*

The parameters a RBFN learns are the centroids, widths, and weights of the RBFs. A fast way to learn centroids and widths is to use a clustering algorithm *e.g. k*-means to obtain *k* RBFs. Then, the weights can be learned using linear or logistic regression (Witten & Frank 2005, Kumar 2005).

Using RBFN to construct a MAU function is treating values of decision attributes as inputs and utilities of alternatives as outputs. For numerical attributes, the values can be directly inputted into a RBFN. For nominal attributes, the values should be transformed to numerical codes before they are inputted to a RBFN. A decision maker provides the holistic evaluations of a set of alternatives to be a training data set and the parameters can be learned by learning algorithms (Lin, Huang, and Yang, 2005).

3   RESEARCH FRAMEWORK

The research framework, illustrated in Figure 2, contains one independent variable, one moderating variable, and several dependent variables about performance. The independent variable is *recommendation method* that can be VSM, SMARTER, or RBFN. The dependent variables include *accuracy*, *time expense*, and *user perceptions*. These variables can be used to measure performance of recommendation or utility construction methods (Lin, Huang, and Yang, 2005). Recommendation accuracy is a measure of whether the recommended items are interesting to the user. Time expense is a measure of how much time the user spends on building his/her MAU function or user profile. This research also investigates user perceptions. Perceived satisfaction, usefulness, and trustworthiness of recommended items and perceived ease of use and comprehensibility of user profile elicitation process are measured. Generally, an item has both nominal and numerical attributes. The different

recommendation approaches have different ways to deal with different scales of attributes. Therefore, this research considers the effect of different item types, item with mostly nominal attributes and item with mostly numerical attributes, which may moderate the effect of recommendation approaches on their performances.
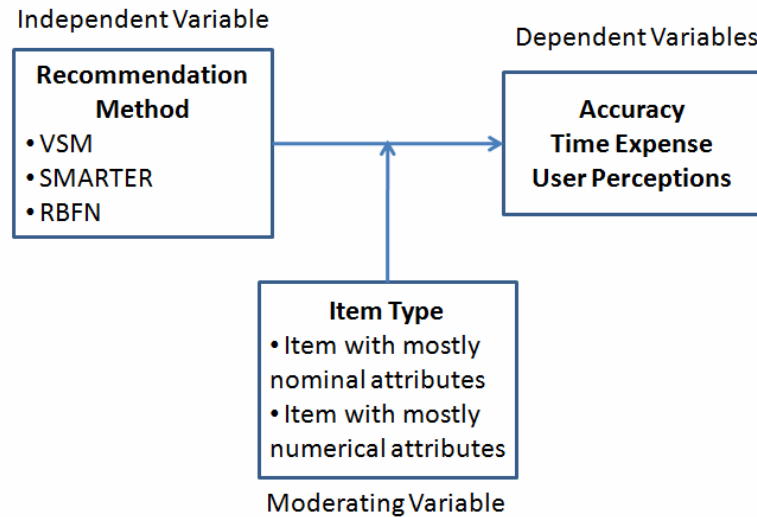

Figure 2.   Research Framework.

The utility-based approaches try to build a MAU function to represent a user's complete information on preferences before recommending items, whereas the traditional content-based approach builds a user profile that may represents partial information on preferences. Therefore, this research expects that RBFN and SMARTER could outperform VSM in terms of accuracy. The hypothesis H1 is developed as follows:

> *H1: A utility-based approach is able to recommend more interesting items to the users than a traditional content-based approach.*

> *H1a: RBFN outperforms VSM in terms of recommendation accuracy.*

> *H1b: SMARTER outperforms VSM in terms of recommendation accuracy.*

Traditional content-based approach asks users to rate items at "like" or "dislike". RBFN method asks users to give each item a utility score. SMARTER method needs to elicit single-dimensional utility functions for all attributes. Therefore, this research expects that users spend more time to build user profiles when using a utility-based approach than using a traditional content-based approach. The hypothesis H2 is derived as follows:

> *H2: A user spends more time to build his/her preference profile when using a utility-based approach than using a traditional content-based approach.*

> *H2a: SMARTER method needs more time expense to build a user preference profile than RBFN and VSM methods do.*

> *H2b: RBFN method needs more time expense to build a user preference profile than VSM does.*

Since different recommendation methods adopt different ways to deal with nominal or numerical attributes. This research also conjectures that different item types could moderate the effects of recommendation methods on their recommendation performances. The hypothesis H3 is proposed as follows:

H3: Recommendation approaches and item types have interaction effects on recommendation performances.

H3a: Recommendation approaches and item types have interaction effects on recommendation accuracy.

H3b: Recommendation approaches and item types have interaction effects on time expense.

H3c: Recommendation approaches and item types have interaction effects on user perceptions.

4   EXPERIMENTAL DESIGN

Two recommendation contexts, recommending movies and recommending notebooks were investigated in this experiment. Movies belong to items with mostly nominal attributes and notebooks belong to items with mostly numerical attributes. The experiment collected 127 data of movies released in the recent year from KingNet (movie.kingnet.com.tw) and collects 81 data of notebooks equipped with dual core processors and Vista operating systems from Yahoo (buy.yahoo.com.tw). We can get the data of movie genre, language, country, director, leading actor, leading actress, company, and revenue ranking from KingNet and get the data of notebook brand, price, processor speed, memory capacity, hard drive capacity, motherboard chipset, video chipset, display size, and weight from Yahoo. Seven undergraduate students were invited to rank the importance of these attributes for renting movie videos and buying notebooks. Finally, this experiment selected five nominal attributes: genre, language, director, leading actor, and leading actress, along with one numeric attribute: revenue ranking for recommending movies; five numerical attributes: price, processor speed, memory capacity, hard drive capacity, and weight, along with one nominal attribute: brand for recommending notebooks.

Three Web-based recommender systems were implemented using VSM, SMARTER, and RBFN methods, respectively. To get user profiles, the VSM recommender system randomly selects a set of items from the database and asks a user to rate each item as "like," "dislike," or "no comment". The system recommends items that are most similar (but not identical) to the items rated "like". If no items are rated "like" the system recommends items that are least similar to the items rated "dislike" to the user.

The SMARTER recommender system firstly constructs single-attribute utility functions for a user. It adopts direct rating method to elicit utility functions of nominal attributes and applies five-point method to elicit utility functions of numerical attributes. For a numerical attribute, the system uses the five points given by a user to train a simple linear regression to form a utility function. After all single-attribute utility functions are built, the system asks the user to rank the importance of each attribute. Attribute weights are calculated using ROC method and the MAU function is built using Formula (3). The items with highest utilities are recommended to the user.

The RBFN recommender system randomly selects a set of items from the database and asks a user to give utility value (between 0 and 100) to each item for providing a set of training examples. The number of RBF in a RBFN could be 2 to $t - 1$, where t is the number of training examples. The system builds all candidate RBFNs using the training examples and calculate their root mean squared errors by t-fold cross-validation. The RBFN with the minimal error is chosen to represent the user's MAU function. According to this function, items with highest utilities are recommended to the user.

4.1   Design of Pilot Test

This research conducts a pilot test to investigate how many training examples are sufficient to train a RBFN for recommendation before executing a laboratory experiment. This pilot test compares the

performances among RBFN with 7 training examples (RBFN7), RBFN with 14 training examples (RBFN14), and RBFN with 21 training examples (RBFN21). Because comparing more than 7±2 items simultaneously is difficult to humans (Miller 1956), the RBFN recommender system in this pilot test asks a subject to give utility values to seven items in each time. Three sets of seven items are evaluated in turn. RBFN7 is trained by the first 7 training data, RBFN14 is trained by the first 14 training data, and RBFN21 is trained by all training data. Three sets of 7 items are recommended to a subject by RBFN7, RBFN14, and RBFN21, respectively. According to the recommended items, the subject is asked to give his/her utility value to each item and the recommendation accuracy is calculated by averaging these utility values. The satisfaction of each set is assessed by giving score using a Likert seven-point scale ranging from -3 to 3. A subject deals with both movie and notebook recommendation contexts and the order of the contexts is randomly determined by the system to eliminate the order effect.

4.2 Design of Laboratory Experiment

In the laboratory experiment, a subject is randomly dispatched to one of the 12 possible paths (see Figure 3). Each path deals with two scenarios each one includes a recommendation context with a recommendation method. This design aims to eliminate order and learning effects. In each scenario the experimental procedure can be divided into the following phases. In phase 1, the recommendation context and item attributes are introduced to the subjects. In phase 2, the recommender system helps subjects to build their user profiles using one of the three approaches. In phase 3, the recommender system recommends 7 items to the subject using corresponding approach.
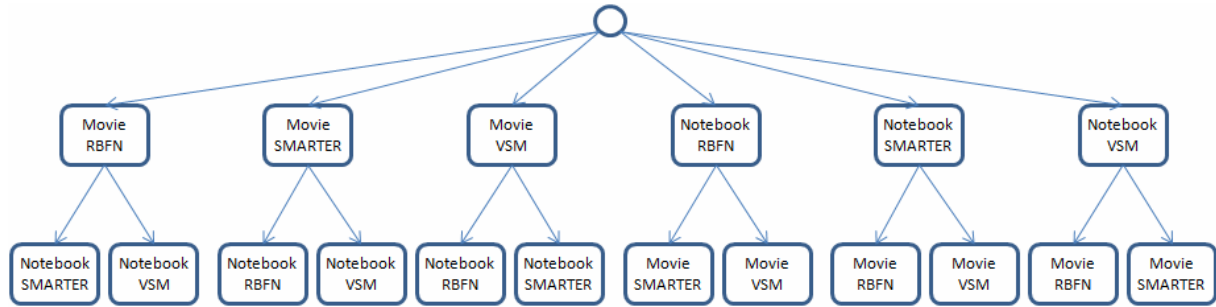

Figure 3.   Experimental Design.

The systems automatically record how much time subjects spend on building their user profiles. To measure recommendation accuracy, subjects are asked to give their utility values to each recommended item and the accuracy is calculated by averaging them. This experiment measures a subject's perceived satisfaction, usefulness, and trustworthiness of recommended results, and perceived ease of use and comprehensibility of user profile elicitation process using Likert seven-point scales.

5 EXPERIMENTAL RESULT

5.1 Result of Pilot Test

There were 9 undergraduate students who majored in Information Systems participated in the pilot test. The test results are depicted in Figure 4 and 5. The results illustrate that 14 training examples is sufficient to train a RBFN for achieving highest accuracy and satisfaction when recommending movies or notebooks. The RBFN recommender system will collect 14 training examples from a

subject for recommending either movies or notebooks in the laboratory experiment. For an impartial comparison, the VSM recommender system will also randomly provide 14 items to a subject to rate for user profile construction in the laboratory experiment.
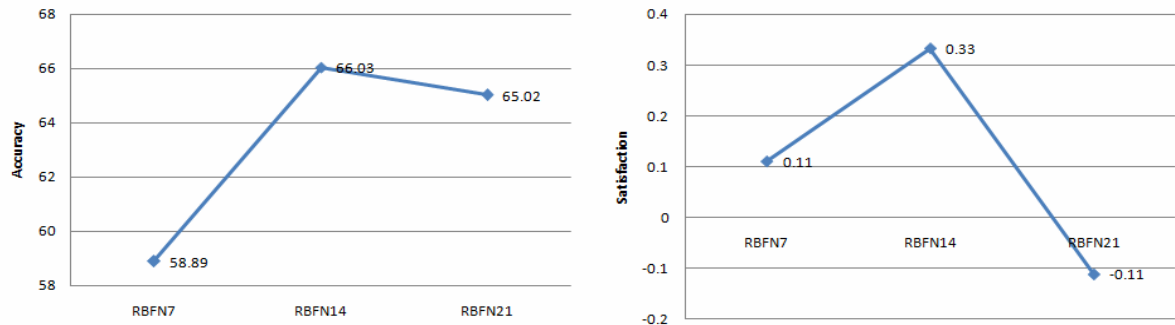

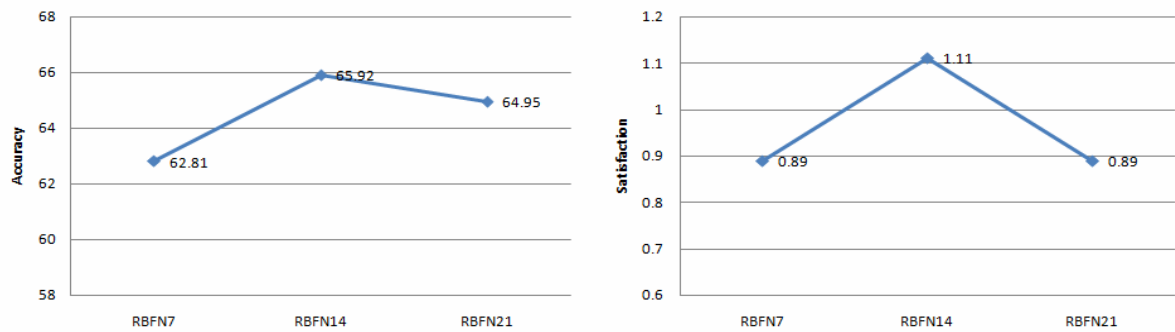*Figure 4.   Performances of movie recommendations in pilot test.*

Figure 5.   Performances of notebook recommendations in pilot test.

5.2   Result of Laboratory Experiment

There were 96 undergraduate students who majored in Information Systems were invited to participate in the laboratory experiment. The ANOVA results shown in Table 1 reveal that the effect of recommendation method on recommendation accuracy is significantly moderated by item type ($F=5.295$, $p<0.01$). The hypothesis H3a is supported. The VSM method significantly gets higher accuracy than the RBFN method when recommending movies ($F=3.226$, $p<0.05$) and the SMARTER method significantly gets higher accuracy than the VSM method when recommending notebooks ($F=3.996$, $p<0.05$). Therefore, the hypothesis H1b is partially supported.

	Accuracy			Time Expense (Seconds)		
	RBFN	SMARTER	VSM	RBFN	SMARTER	VSM
Item with mostly nominal attributes	67.97 (10.90) / 33	73.35 (12.27) / 28	74.25 (9.57) / 35	130.00 (93.12) / 33	1042.64 (326.49) / 28	79.80 (34.33) / 35
Item with mostly numerical attributes	68.85 (13.97) / 36	71.71 (10.07) / 31	60.57 (21.89) / 29	180.03 (100.12) / 36	324.19 (107.40) / 31	107.76 (85.41) / 29
	$F=5.295, p=0.006$**			$F=132.827, p=0.000$**		

Mean (Standard Deviation) / Number of Subjects.
** The interaction effect is significant at the 0.01 level.

Table 1.   Effects of recommendation method and item type on accuracy and time expense.

The ANOVA results also show that the recommendation method and item type have a significant interaction effect on time expense ($F=132.827$, $p<0.01$). SMARTER method costs significantly more time when dealing with movie recommendation than notebook recommendation. The hypothesis H3b is supported. The SMARTER method needs significantly more time to elicit a user profile than the RBFN and VSM methods do for recommending movies ($F=254.808$, $p<0.01$) and for recommending notebooks ($F=38.035$, $p<0.01$). The RBFN method also significantly takes more time to construct a user profile than the VSM method for recommending movies and notebooks ($p<0.05$). Therefore, the hypothesis H2 is supported.

Subjects' perceptions of each method were assessed by their perceived satisfaction, usefulness, and trustworthiness of recommended items; and perceived ease of use and comprehensibility of user profile elicitation process. The results (see Table 2 and Table 3) show that there exist no significant interaction effects between recommendation method and item type on subjects' perceptions. The hypothesis H3c is not supported. Notably, subjects felt that the utility elicitation process of the RBFN method in notebook recommendation context had lower comprehensibility than the utility elicitation process of the SMARTER method ($F=3.194$, $p<0.05$).

	Satisfaction			Usefulness			Trustworthiness		
	RBFN	SMARTER	VSM	RBFN	SMARTER	VSM	RBFN	SMARTER	VSM
Item with mostly nominal attributes	0.73 (1.59) / 33	1.43 (1.29) / 28	0.97 (1.15) / 35	0.39 (1.82) / 33	0.82 (1.63) / 28	0.69 (1.18) / 35	0.52 (1.54) / 33	0.96 (1.48) / 28	0.57 (1.14) / 35
Item with mostly numerical attributes	0.28 (1.49) / 36	0.61 (1.52) / 31	0.03 (1.59) / 29	0.17 (1.30) / 36	0.55 (1.63) / 31	0.17 (1.67) / 29	0.17 (1.25) / 36	0.52 (1.59) / 31	0.07 (1.51) / 29
	F=0.513, p=0.599			F=0.160, p=0.852			F=0.050, p=0.951		

Mean (Standard Deviation) / Number of Subjects.

Table 2.   *Effects of recommendation method and item type on perceived satisfaction, usefulness, and trustworthiness of recommended items.*

	Ease of Use			Comprehensibility		
	RBFN	SMARTER	VSM	RBFN	SMARTER	VSM
Item with mostly nominal attributes	1.00 (1.87) / 33	0.64 (1.89) / 28	0.91 (1.34) / 35	0.76 (1.70) / 33	1.07 (1.54) / 28	0.80 (1.47) / 35
Item with mostly numerical attributes	0.06 (1.67) / 36	0.94 (1.73) / 31	0.38 (1.93) / 29	-0.11 (1.60) / 36	0.87 (1.54) / 31	0.03 (1.90) / 29
	F=2.055, p=0.131			F=0.754, p=0.472		

Mean (Standard Deviation) / Number of Subjects.

Table 3.   *Effects of recommendation method and item type on perceived ease of use and comprehensibility of user profile elicitation process.*

5.3   Discussion

The experiment found that the VSM method outperforms utility-based methods in terms of recommendation accuracy and time expense when recommending items with mostly nominal

attributes. But the VSM method has poor accuracy when recommending items with mostly numerical attributes. This consequence is possibly caused by trade-offs existing among notebook attributes *e.g.* higher processor speed, memory capacity, hard drive capacity, and lower weight usually come with higher price. Both notebooks with better equipments and higher prices and notebooks with worse equipments and lower prices can satisfy subjects. The trade-offs made subjects feel difficult to provide overall evaluations of alternatives, and recommending items those are similar to past liked items cannot cover subjects' complete preferences (overspecialization problem).

The SMARTER method comparatively has good recommendation accuracy especially in the notebook recommendation context. But the SMARTER method costs most time to elicit user profiles especially in the movie recommendation context. Using a decomposed method to construct subjects' MAU functions make they feel easy to comprehend the elicitation processes and tend to achieve higher recommendation accuracy. However, the tedious elicitation processes cost much time particularly when using direct rating method to rate many values of nominal attributes.

The recommendation accuracy and time expense of the RBFN method are reliable in different recommendation contexts. Its accuracy is steadily about 68 and its time expense on utility elicitation process has little fluctuation within 1 minute. However, the holistic approach makes the elicitation processes harder to be comprehended when recommending items with trade-off attributes.

6   CONCLUSION

Utility-based recommendation methods try to model a user's multi-attribute utility function and recommend items with highest utilities based on this function. This research compared different utility-based recommendation methods including the RBFN and SMARTER methods with traditional content-based method, the VSM method, in terms of recommendation accuracy, time expense, and user perceptions in contexts of recommending different types of items. A laboratory experiment was conducted and found that recommendation method and item type has interaction effect on recommendation accuracy and time expense on elicitation of user profile. The VSM method is suitable to recommend items with mostly nominal attributes, and the SMARTER method is suitable to recommend items with mostly numerical attributes. The RBFN method has reliable accuracy and time expense no matter the types of items. This study expects the research results can help developers of recommender systems to design and choose suitable recommendation methods. Future researches could investigate long-term performances of different recommendation methods and the moderating effects of more various types of items should be examined.

Acknowledgment

References

Adomavicius, G. and Tuzhilin, A. (2005). 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, 17 (6), 734-749.

Balabanovic, M. and Shoham, Y. (1997). Fab: Content-based collaborative recommendation. Communications of the ACM, 40 (3), 66-72.

Edwards, W. (1977). How to use multi-attribute utility measurement for social decision making. IEEE Transactions on System, Man and Cybernetics, 3 (5), 326-340.

Edwards, W. and Barron, H.F. (1994). SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement. Organizational Behavior and Human Decision Processes, 60, 306-325.

Keeny, R. and Raiffa, H. (1976). Decision with Multiple Objectives: Preferences and Value Tradeoffs. John Willey & Sons.

Kumar, S. (2005). Neural Networks: A Classroom Approach. McGraw-Hill.

Laskey, K.B. & Fischer, G.W. (1987). Estimating utility functions in the presence of response error. Management Science, 33 (8), 965-980.

Lin, F.-R., Huang, S.-L., and Yang, Y.-C. (2005). Using radial basis function networks to model multi-attribute utility functions. In Proceedings of the 4th Workshop on e-Business, Las Vegas, USA.

Malakooti, B. and Zhou, Y.Q. (1994). Feedforward artificial neural networks for solving discrete multiple criteria decision making problems. Management Science, 40 (11), 1542-1561.

Miller, G.A. (1956). The magical number seven plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, 81-97.

Pomerol, J.-C. and Barba-Romero, S. (2000). Multicriterion Decision in Management: Principles and Practice. Kluwer Academic Publishers.

Resnick, P. and Varian, H.R. (1997). Recommender systems. Communications of the ACM, 40 (3), 56-58.

Salton, G. (1989). Automatic Text Processing. Addison-Wesley.

Saaty, T.L. (1980). The Analytic Hierarchy Process. New York: McGraw-Hill.

Schoemaker, P.J.H. and Waid, C.C. (1982). An experiment comparison of different approaches to determining weights in additive utility models. Management Science, 28 (2), 182-196.

Srivastava, J., Connolly, T., and Beach, L.R. (1995). Do rank suffice? A comparison of alternative weighting approaches in value elicitation. Organization Behavior and Human Performance, 63, 112-116.

Sun, M., Stam, A., and Steuer, R.E. (1996). Solving multiple objective programming problems using feed-forward artificial neural networks: The interactive FFANN procedure. Management Science, 42 (6), 835-849.

Wang, M.S. and Yang, J.S. (1998). A multi-criterion experimental comparison of three multi-attribute weight measurement methods. Journal of Multi-Criterion Decision Analysis, 7, 340-350.

Witten, I.H. and Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques. 2nd Edition. Morgan Kaufmann.

Yamamoto, N., Saito, M., Miyazaki, M., and Koike, H. (2005). Recommendation algorithm focused on individual viewpoints. In Proceedings of the 2nd IEEE Consumer communications and Networking Conference, pp. 5-70, Las Vegas, USA.

Yap, C.S., Raman, K.S., and Leong, C.M. (1992). Methods for information system project selection: an experimental study of AHP and SMART. In Proceedings of the 25th Hawaii International Conference on System Sciences, pp. 578-589, Hawaii, USA.