A Hybrid Recommendation Approach for Hierarchical Items

Dianshuang Wu [a,b]
[a] School of Automation
Northwestern Polytechnical University
Xi'an, Shaanxi 710072, P.R. China
wds@mail.nwpu.edu.cn

Jie Lu [b], Guangquan Zhang [b]
[b] Decision Systems & e-Service Intelligence (DeSI) Lab
Centre for Quantum Computation & Intelligent Systems
Faculty of Engineering and Information Technology,
University of Technology, Sydney
P.O. Box 123, Broadway, NSW 2007, Australia
jielu@it.uts.edu.au, zhangg@it.uts.edu.au

Abstract—**Recommender systems aim to recommend items that are likely to be of interest to the user. In many business situations, complex items are described by hierarchical tree structures, which contain rich semantic information. To recommend hierarchical items accurately, the semantic information of the hierarchical tree structures must be considered comprehensively. In this study, a new hybrid recommendation approach for complex hierarchical tree structured items is proposed. In this approach, a comprehensive semantic similarity measure model for hierarchical tree structured items is developed. It is integrated with the traditional item-based collaborative filtering approach to generate recommendations.**

Keywords-hierarchical items; tree similarity measuring; recommender systems

I. INTRODUCTION

With the advent of large-scale E-commerce applications, personalization has gained momentum as a means of challenging the information overload, as well as of understanding, and catering to, the needs of individuals or groups of customers [1]. A recommender system is one of the most popular applications of personalization techniques, aiming at filtering out the uninterested items or predicting the interested ones automatically on behalf of the users according to their personal preferences [2]. Recommender systems have gained much attention from both industry and academia since their appearance [3].

Most existing recommender systems employ collaborative filtering (CF) techniques to suggest items of which a particular user is interested in [4]. However, traditional CF techniques don't understand items' semantic features, which can result in no guarantee of prediction accuracy for the complex items. Thus, a hybrid recommendation method integrating the semantic similarity between items is used for improving the accuracy of recommendations [1, 2, 4, 5]. Many models and algorithms have been developed to measure the semantic similarity between two items, which are described in a set of attributes. However, in many business situations, products or services are so complex that they can only be described by hierarchical tree structures.

Fig. 1 shows two examples of telecom service packages. Obviously, they are hierarchical items and viewed as tree structures. Taking the first package as an example, the whole package is composed of three sub-services: fixed line, mobile, and broadband services. Each sub-service is described by several aspects in details. For example, the broadband service is described by four aspects: price, contract duration, allowance, and throttle speed. From the price aspect, we can see that the price of broadband service is $40 per month. From the tree structure, it can be also seen that every node is assigned a weight to reflect its importance degree to its siblings. To recommend these hierarchical items accurately, the semantic information of hierarchical tree structures must be considered. On the other hand, the difference between two telecom service packages in Fig. 1 shows that the tree structures, nodes' weights, terms and values of different hierarchical items are all different. This makes measuring the semantic similarity between these hierarchical tree structured items not trivial. In this study, a comprehensive similarity measure model to evaluate the semantic similarity between hierarchical tree structured items is proposed. A hybrid recommendation approach integrating the similarity measure model with the traditional item-based CF techniques is presented.

The rest of this paper is organized as follows. Section II presents the related works on this study. Section III describes the features of hierarchical items formally. A comprehensive semantic similarity measure model for hierarchical items is provided in Section IV. In Section V, a hybrid recommendation approach which integrates the semantic similarity measure model with the traditional item-based CF is presented. A numerical example is also given. Finally, conclusions and future study are discussed in Section VI.

II. RELATED WORKS

Various approaches for recommender systems have been developed. It has been proven that the CF recommendation approach is the most successful and widely used approach for recommendation systems [6]. CF approach recommends items for a particular user using the opinions of other people based on the assumption that people with similar tastes will rate things similarly [7]. It can be further divided into user-based and item-

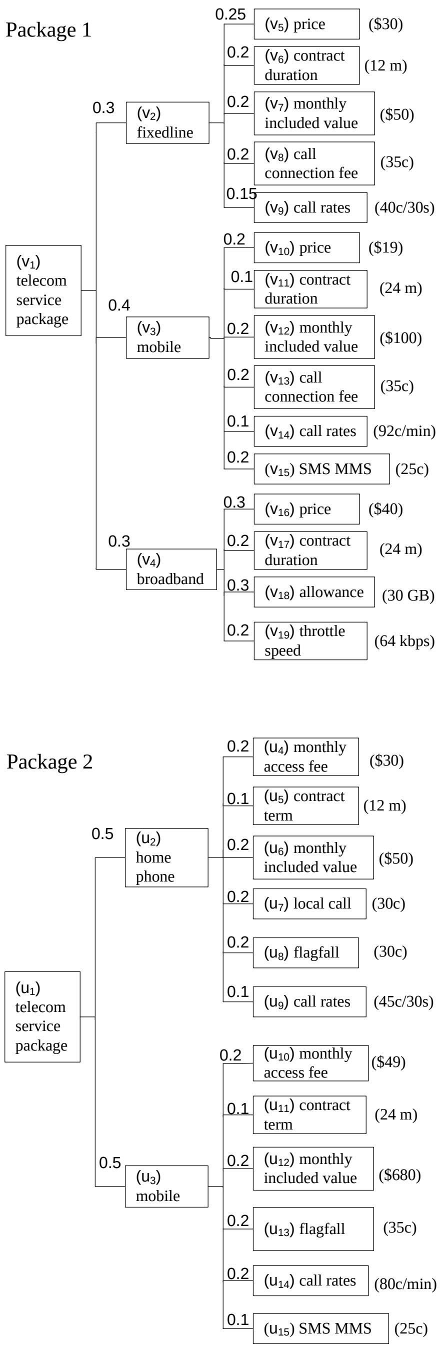

Figure 1. Two examples of telecom service package

based CF approaches [8]. User-based CF first finds a set of nearest neighbors of a target user by computing correlations or similarities between users. The prediction value of unknown items to the target user is then computed according to his/her nearest users. In contrast, item-based CF attempts to find a set of similar items that are rated by different users in some similar way. Then, for a target item, predictions can be generated, for example, by taking a weighted average of the active user's item ratings on these neighbor items. Item-based CF has been shown to achieve prediction accuracies that are comparable to or even better than user-based CF algorithms [9]. The basic assumption in CF approaches is that there are sufficient historical data for measuring similarity between items. However, this assumption does not hold in various application domains. Therefore, CF approaches have sparsity and cold-start problems.

To improve the prediction accuracy, hybrid recommender systems which integrate other information sources with CF approaches are developed. Because the computation of item similarities is independent of the methods used for generating predictions, multiple knowledge sources, including structured semantic information about items, can be brought to bear in determining similarities among items. The integration of semantic similarities for items allows the system to make inferences based on the underlying reasons for which a user may or may not be interested in a particular item [10]. In [5], items are mapped to a set of concepts which are the nodes of domain ontology. A semantic match with the domain ontology is used to find similar users. A semantic product relevance model is integrated into the traditional item-based CF recommender system in [8] to recommend the one-and-only items, such as trade exhibitions. In [4], they integrate fuzzy sets based semantic similarity and traditional item-based CF methods to improve recommendation accuracy. In these approaches, items are all represented by sets of attributes or concepts. To the best of our knowledge, there is no work to integrate the semantic similarity measure on complex hierarchical tree structured items into the recommender systems.

III. HIERARCHICAL ITEM TREES

To describe the features of hierarchical items formally, a hierarchical item tree (HI-tree) is defined as follows.

Definition 1: HI-tree. An HI-tree is a structure $T=(V, E, A, W, R)$, in which V is a finite set of nodes, E is a binary relation on V where each pair $(u, v) \in E$ represents the parent-child relationship between two nodes $u, v \in V$, A is a set of attributes assigned to each node in V, W is a function to assign each node a weight to represent its importance degree to its siblings, thereby satisfying the sum of the weights of all children of one node is 1, and R is a function to assign a value to every leaf node to describe the relevant attribute.

The two telecom service packages in Fig. 1 are two HI-trees. Taking Package 1 as an example, it has nineteen nodes, and every node is assigned an attribute. The number beside the edge is the weight of the child. The data in the bracket beside the leaf nodes are their values.

Two features of the HI-tree should be highlighted: (1) every node in the HI-tree is associated with a concept, which is obtained from the attributes. As a hierarchical structure, the concept of one node depends, not only on the attribute itself, but also its children's; (2) every node in the HI-tree has a value. The leaves' values are indicated by R, and the internal nodes' values can be computed by aggregating their children's.

As the attribute terms in different HI-trees are probably different. To evaluate the conceptual correspondence between attributes in different HI-trees, a conceptual similarity measure between attributes is introduced as in [11].

Definition 2: Attribute Conceptual Similarity Measure. An attribute conceptual similarity measure sc_{A_1,A_2} is a set of mappings from two attribute sets A_1, A_2 used in different HI-trees to the set [0, 1], $sc_{A_1,A_2}: A_1 \times A_2 \rightarrow [0, 1]$, in which each mapping denotes the conceptual similarity between two attributes. For convenience, the subscript A_1, A_2 is omitted if there is no confusion. For $a_1 \in A_1$, and $a_2 \in A_2$, we say a_1 and a_2 are similar if $sc(a_1, a_2)>0$, and the larger $sc(a_1, a_2)$ is, the more similar the two attributes are.

Conceptual similarity between two attributes can be given by domain experts or calculated based on linguistic analysis methods. As an example, we define the conceptual similarity between the attributes of two HI-trees in Fig. 1 as follows:

sc(*fixed line, home phone*)*=0.9, sc*(*price, monthly access fee*)*=0.8, sc*(*contract duration, contract term*)*=1, sc*(*call connection fee, flag fall*)*=0.9.*

IV. Semantic similarity measure and algorithms for HI-trees

A semantic similarity measure model for HI-trees is presented in this section. In this model, a maximum correspondence tree mapping is constructed to identify the corresponding node pairs of two HI-trees based on nodes' structures and concepts, and the conceptual similarity between two HI-trees is evaluated. Based on the mapping, the value similarity between two HI-trees is evaluated, and the final similarity measure between two HI-trees is assessed as a weighted sum of their conceptual and value similarities.

A. Maximum correspondence tree mapping

Corresponding nodes in different HI-trees are identified based on both their structures and concepts.

As the hierarchical features of the HI-trees, there are two structural restrictions: (1) the corresponding nodes in the mapping should be at the same depth; (2) two separate sub-trees in one tree should be mapped to two separate sub-trees in another.

In addition to satisfying structural restrictions, it is important that the corresponding nodes have a high conceptual similarity degree. Therefore, concept correspondence degree between nodes in different trees should be evaluated.

Given two nodes v and u in two HI-trees, there are three situations: (1) both v and u are leaves, (2) v is a leaf and u is an internal node, (3) both v and u are internal nodes. In the first case, the concept correspondence degree between v and u can be evaluated by the conceptual similarity of their attributes. In the other two cases, as the internal node's concept is also affected by its children, the children's concepts should be considered. Therefore, the concept correspondence degree should be computed recursively in the latter two cases. The computation process of the concept correspondence degree between v and u, denoted by $cord(v,u)$, is shown in Algorithm 1. In the algorithm, $v.a$ and $u.a$ represent attributes of v and u respectively, w_{1k} and w_{2l} represent the weights of v_k and u_l respectively, and α represents the influence factor of the parent node.

Algorithm 1. Concept correspondence degree computation algorithm

```
cord(v, u, B)
input: two nodes v and u
output: concept correspondence degree between v and u
1  if both v and u are leaves
2    return sc(v.a, u.a)
3  else if u is an internal node, and u_1, u_2,…, u_q be u's children
4    return $\alpha \cdot sc(v.a,u.a) + (1-\alpha)\cdot \sum_{i=1}^{q} w_{2i} \cdot cord(v,u_i,B)$
5  else
6    $C(v) \leftarrow$ v's children v_1, v_2,…, v_p
7    $C(u) \leftarrow$ u's children u_1, u_2,…, u_q
8    for i=1 to p
9      for j=1 to q
10       $c_{ij} \leftarrow cord(v_i, u_j, B)$
11   $m \leftarrow ComputeMatching(C(v) \cup C(u), c)$
12   for each $(v_k, u_l) \in m$, if $c_{kl} > 0$
13     $B(v_k) \leftarrow B(v_k) \cup \{u_l\}$
14   return $\alpha \cdot sc(v.a,u.a) + (1-\alpha)\cdot \sum_{(v_k,u_l)\in m} ((w_{1k}+w_{2l})/2)\cdot c_{kl}$
```

When both v and u are internal nodes (lines 5-14), to fully consider the influence of their children's concepts, a bipartite graph $G_{v,u}$ is constructed with v and u's children as follows: $V=C(v) \cup C(u)$, $E=\{(s,t):s \in C(v),\ t \in C(u)\}$. The weight of edge (s,t) is defined as the concept correspondence degree between s and t. To identify the maximum corresponding node pairs between $C(v)$ and $C(u)$, a maximum weighted bipartite matching (MWBM) problem of $G_{v,u}$ need to be resolved. The function $ComputeMatching(\cdot)$ [12] returns the MWBM of $G_{v,u}$.

From the recursive computation process of node concept correspondence degree, it is obvious that $cord(root(T_1),root(T_2))$ is computed by aggregating the concept correspondence degree of all corresponding node pairs, which reflects the conceptual similarity between two HI-trees. Thus, the conceptual similarity between two HI-trees T_1 and T_2 is defined as:

$$sct(T_1, T_2)= cord(root(T_1),root(T_2)) \qquad (1)$$

In Algorithm 1, B is a node set list which is used to record the results of MWBM problems. During the computation process of $cord(root(T_1),root(T_2))$, all the MWBMs are recorded in B. As proved in [13], there is a unique maximum correspondence tree mapping $M \subseteq V_1 \times V_2$ so that $M \subseteq B$. Given B, the corresponding maximum correspondence tree mapping M between T_1 and T_2 can be constructed as follows: Set $M(root(T_1))$ to $root(T_2)$ and, for all nodes $v \in V_1$ in pre-order, set $M(v)$ to the unique node u with $(v,u) \in B$ and $(parent(v),parent(u)) \in B$.

B. HI-tree value similarity measure

Based on the maximum correspondence tree mapping M between two HI-trees T_1 and T_2, the value similarity between T_1 and T_2 can be computed.

The value similarity between two corresponding nodes in *M* is evaluated first. As only leaf nodes are assigned values in HI-trees initially, for any $(v,u) \in M$, there are two cases: (1) *v* is a leaf node, or none of *v*'s children are in *M*, (2) some of *v*'s children are in *M*. We provide the computation formulas of the value similarity between *v* and *u*, $sv_M(v,u)$ for the two cases respectively.

In case 1,

$$sv_M(v,u)=s(value(v),value(u)) \tag{2}$$

where *value*(*v*) denotes *v*'s value and $s(\cdot)$ denotes a value similarity measure. If *v* is a leaf node, *value*(*v*) is assigned initially. Otherwise, it is computed by aggregating its children's values. $s(\cdot)$ can be defined according to the specific applications.

In case 2, let $v_1, v_2,\ldots, v_p$ be *v*'s children and $u_1, u_2,\ldots, u_q$ be *u*'s children.

$$sv_M(v,u)=\sum\nolimits_{(v_i,u_j)\in M} sv_M(v_i,u_j)\cdot(w_{1i}+w_{2j})/2 \tag{3}$$

where w_{1i} is the weight of v_i in T_1 and w_{2j} is the weight of u_j in T_2.

As the recursive characteristic of the computation of value similarity between nodes, the value similarity between the roots of two HI-trees is computed by aggregating the value similarity of all corresponding node pairs, which represents the value similarity of the two HI-trees. Thus, the value similarity between two HI-trees T_1 and T_2 is defined as:

$$svt(T_1, T_2)=sv_M(root(T_1),root(T_2)) \tag{4}$$

C. Semantic similarity measure for HI-trees

Based on the conceptual similarity and value similarity of two HI-trees, the semantic similarity measure of two HI-trees T_1 and T_2 is defined as follows:

$$S_{sem}(T_1, T_2)=\alpha_1\cdot sct(T_1, T_2)+\alpha_2\cdot svt(T_1, T_2) \tag{5}$$

where $\alpha_1+\alpha_2=1$.

V. A HYBRID RECOMMENDATION APPROACH

The proposed hybrid recommendation approach looks into the set of items the target user has rated and computes how similar they are to the target item *i* and then selects *k* most similar items $\{ i_1, i_2,\ldots, i_k \}$. To evaluate the similarity between two items, our approach first calculates the item-based CF similarity and semantic similarity between them respectively, and then combines the two similarities to get the total similarity between them. Once the most similar items to the target item are found, the prediction is then computed by taking a weighted average of the target user's ratings on these similar items.

A. Recommendation generation

The proposed approach generates recommendations through the following four steps.

Step 1: Computing item-based CF similarity

This step computes the similarity of two items *i* and *j* according to users' ratings on items. Many measures can be used to compute the similarity between items. In our approach, the adjusted cosine similarity measure [10] is employed in order to take into account the variances in user ratings. Let U_{ij} be the set of users who rated the items *i* and *j* together. The CF similarity between items *i* and *j*, $S_{CF}(i,j)$ is given in the following formula:

$$S_{CF}(i,j)=\frac{\sum_{u\in U_{ij}}(r_{u,i}-\overline{r_u})\times(r_{u,j}-\overline{r_u})}{\sqrt{\sum_{u\in U_{ij}}(r_{u,i}-\overline{r_u})^2}\times\sqrt{\sum_{u\in U_{ij}}(r_{u,j}-\overline{r_u})^2}} \tag{6}$$

where $r_{u,i}$ and $r_{u,j}$ are user *u*'s ratings on item *i* and *j* respectively, and $\overline{r_u}$ is the average ratings of all items which have been rated by user *u*.

Step 2: Computing semantic similarity

Based on the similarity measure model for hierarchical items presented in Section IV, the semantic similarity between items *i* and *j*, $S_{sem}(i,j)$ is computed by (5).

Step 3: Integrating semantic similarity with item-based CF similarity

In this step, the total similarity between items *i* and *j*, *TotalSim*(*i*,*j*) is computed by integrating the two similarity measures computed in the last two steps. As $S_{CF}(i,j)$'s value range is [-1, 1] and $S_{sem}(i,j)$'s is [0, 1], the $S_{CF}(i,j)$ is transformed first as follows: $S_{CF}'(i,j)=(1+S_{CF}(i,j))/2$. *TotalSim*(*i*,*j*) is computed by the formula:

$$TotalSim(i,j)=\beta\times S_{CF}'(i,j)+(1-\beta)\times S_{sem}(i,j) \tag{7}$$

where β is a semantic combination parameter specifying the weight of similarity in the integrated measure. Finding an appropriate value for β is usually highly dependent on the characteristics of the data used. We can choose a proper value by performing sensitivity analysis for particular data sets.

Step 4: Generating recommendations

After computing the similarity between items, we select *k* most similar items to the target item and generate a predicted value for the target item in this step. Let S_i be the set of most similar items to the target item *i*, the prediction on the item *i* for a target user *u*, $P_{u,i}$ is computed by the following formula:

$$P_{u,i}=\frac{\sum_{n\in S_i} r_{u,n}\times TotalSim(i,n)}{\sum_{n\in S_i} TotalSim(i,n)} \tag{8}$$

where $r_{u,n}$ denotes user *u*'s rating on item *n*. *TotalSim*(*i*,*n*) shows the similarity weight between item *n* and item *i*.

B. An illustrative scenario

An example is given to illustrate the proposed approach. We assume that there are five telecom service packages (Package 1 to Package 5) available. The first two packages are illustrated in Fig. 1. There are four users (User 1 to User 4). The scale from '1' to '5' indicates the user's interest to the products. A user-item rating matrix is depicted in Table I. It can be seen that package 5 is a new product.

Step 1: Computing item-based CF similarity.

The item-based CF similarity between items can be calculated by (6), and therefore we have the item-based CF similarity matrix shown in Table II. As Package 5 is a new product, there is no rating data available. Therefore, the item-based CF similarities between Package 5 with others can't be computed.

Step 2: Computing semantic similarity

Based on the semantic similarity measure model for hierarchical items presented in Section IV, the semantic similarities between items are computed. Taking Package 1 and 2 as an example, let the coefficients α , α_1 and α_2 be all 0.5. The conceptual and value similarity between them can be computed respectively as: $sct(P_1,P_2)=0.892$ and $svt(P_1,P_2)=0.562$. The final semantic similarity between them can be computed by (5), and $S_{sem}(P_1,P_2)=0.73$. The semantic similarity matrix is shown in Table III.

Step 3: Computing the total similarity

Let β be 0.4, the total similarity between these products can be computed by (7), which is shown in Table IV.

Step 4: Generating recommendations.

Based on the total similarity between products computed in the last step, the predictions can be calculated by (8):

$P_{1,3}=3.8$, $P_{1,5}=4.4$, $P_{2,4}=3.3$, $P_{2,5}=3.9$, $P_{3,2}=3.1$, $P_{3,5}=2.9$, $P_{4,1}=3.8$, $P_{4,5}=3.2$, where P_{ij} denotes the prediction on the telecom service package *j* for user *i*.

TABLE I. USER-ITEM RATING MATRIX

	Telecom service package				
	Package 1	***Package 2***	***Package 3***	***Package 4***	***Package 5***
User 1	5	5		3	
User 2	5	4	2		
User 3	4		1	3	
User 4		5	2	2	

TABLE II. ITEM-BASED CF SIMILARITY MATRIX

S_{CF}	***Package 2***	***Package 3***	***Package 4***	***Package 5***
Package 1	0.8	-1	-0.22	
Package 2		-0.65	-0.82	
Package 3			0.22	
Package 4				

TABLE III. SEMANTIC SIMILARITY MATRIX

S_{sem}	***Package 2***	***Package 3***	***Package 4***	***Package 5***
Package 1	0.73	0.22	0.4	0.7
Package 2		0.43	0.3	0.6
Package 3			0.66	0.33
Package 4				0.42

TABLE IV. TOTAL SIMILARITY MATRIX

TotalSim	***Package 2***	***Package 3***	***Package 4***	***Package 5***
Package 1	0.798	0.132	0.396	0.62
Package 2		0.328	0.216	0.56
Package 3			0.64	0.398
Package 4				0.452

The example shows that the semantic similarity between hierarchical tree structured items is evaluated comprehensively in the proposed approach. The semantic similarity between items provides rich information to generate recommendations, which will make the recommendations more accurate. The hybrid approach can also recommend new items, such as Package 5, which solves the cold-start problem in traditional CF approaches.

VI. CONCLUSION AND FUTURE WORK

This study proposes a hybrid recommendation approach for complex hierarchical tree structured items. In the approach, a comprehensive semantic similarity measure model for hierarchical tree structured items is developed. It is integrated with the traditional item-based CF approach to generate recommendations. The proposed hybrid recommendation approach can fully utilize the semantic features of items and it can also solve the sparsity and cold-start problems in the traditional item-based CF approach. Further study includes developing software based on the proposed hybrid recommendation approach and experimentally evaluating the approach.

ACKNOWLEDGMENTS

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Project DP0880739 and the China Scholarship Council (No. 2009629052).

REFERENCES

[1] C. Cornelis, J. Lu, X.. Guo, and G. Zhang, (2007) "One-and-only item recommendation with fuzzy logic techniques," *Information Science*, vol. 177, pp. 4906-4921.

[2] C. Porcel, A. G. López-Herrera, and E. Herrera-Viedma, (2009) "A recommender system for research resources based on fuzzy linguistic modeling," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5173-5183.

[3] G. Adomavicius and A. Tuzhilin, (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*, vol. 17, no.6, pp. 734-749.

[4] J. Lu, Q. Shambour, Y. Xu, Q. Lin, and G. Zhang, (2010) "A hybrid semantic recommendation system for personalized government-to-business e-service," accepted by *Internet Research*.

[5] J. Lee, K. Nam, and S. Lee, (2009) "Semantics based collaborative filtering," R. Lee, N. Ishii (Eds.): *Software Engineering, Artificial Intelligence, SCI 209*, pp. 201-208.

[6] Z. Huang, D. Zeng, and H. Chen, (2007) "A comparison of collaborative-filtering recommendation algorithms for e-commerce", *IEEE Intelligent Systems*, vol. 22, no. 5, pp. 68-78.

[7] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, (2007) "Collaborative filtering recommender systems," P. Brusilovsky, A. Kobsa, and W. Nejdl (Eds.): *The Adaptive Web, LNCS 4321*, pp. 291-324.

[8] X. Guo, and J. Lu, (2007) "Intelligent e-government services with personalized recommendation techniques," on the special issue on *E-service Intelligence of International Journal of Intelligent Systems*, vol. 22, no. 5, pp. 401-417.

[9] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, (2001) "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th international conference on World Wide Web* ,pp. 285-295

[10] B. Mobasher, X. Jin, and Y. Zhou, (2003) "Semantically enhanced collaborative filtering on the web," *Web Mining: FromWeb to SemanticWeb In Proceedings of the 1st European Web Mining Forum (EWMF2003)*, pp. 57-76.

[11] Y. Xue, C. Wang, H.H. Ghenniwa, and W. Shen, (2009) "A new tree similarity measuring method and its application to ontology comparison," *Journal of Universal Computer Science*, vol. 15, no. 9, pp. 1766-1781.

[12] D. Jungnickel, (2008). *Graphs, networks, and algorithms*, Berlin: Springer, pp. 419-430.

[13] G. Valiente, (2002). *Algorithms on trees and graphs*, New York: Springer, pp. 206-224.