A New Ontology-Based User Modeling Method for Personalized Recommendation

Jiangling Yuan, Hui Zhang, Jiangfeng Ni
State Key Laboratory of Software Development Environment
Beihang University, School of Computer Science
100191, Beijing, China
{jalunnier, hzhang, nijf}@nlsde.buaa.edu.cn

Abstract—Personalized recommendation is an effective method to resolve the current problem of Internet information overload. In the recommendation systems, user modeling is a crucial step. Whether the model can accurately describe the users’ interests directly determines the quality of the personalized recommendations. At present in most personalized service systems keywords models or user-item models are used to describe the users’ preferences, but vectors or matrixes used in these models do not contain semantic information, so it is difficult to accurately model the users’ interests and hobbies, and it is also hard to extend the users’ interests. Ontology as a tool used to describe the domain knowledge is very powerful in conceptual describing and logical reasoning. Computation of the neighbor set of users or resources is also an important step in the recommendation, but at present three commonly used similarity algorithms have some shortcomings which lead the system sometimes difficulty to find similar users or resources. This paper presents a new ontology-based user modeling approach and an improved similarity algorithm. Our experiments show that the user model presented in this paper can effectively describe the users’ personalized preferences, and we also prove that the improved similarity algorithm is better than other three commonly used similarity algorithms.

Keywords—personalized recommendation; ontology; semantic reasoning; user modeling; similarity measure

I. INTRODUCTION

The explosive growth in the information available on the Web has prompted the need for developing Web personalization systems that understand and exploit user preferences to dynamically serve customized content to individual users [1]. The method that how to build the user model determines the model whether can accurately describe the users’ real interests and the system whether can recommend the right items to users, so user modeling has become the key step in the personalized recommendation systems [2-3]. At present Most of the personalized recommendation systems use keywords vectors or user-resource matrix to represent the users’ interests. However, with the increase of users and resources in system, the scale of vectors or matrixes will tremendously grow, which drops the efficiency of the system. As we all know there are semantic relationships between the resources visited by users, but some commonly used models haven’t taken advantage of these semantic relationships, some simply make use of the semantic information, so these models can't accurately describe the users’ interests [4-6]. Ontology is used to depict the domain knowledge, provides the common understanding of the knowledge about one area, defines the common cognitive vocabulary, and gives the clear definition on different domain terms. This paper presents a new ontology-based user modeling method which uses ontology concept hierarchy tree to represent the users’ interests, and we use the reasoning and extension technique of the ontology to mine the users’ potential interests. Experiment results show that this method can more accurately describe the users’ interests.

In the recommendation systems similarity measure plays an important role, which is the base procedure for finding out the neighbor set of users or resources. At present three commonly used similarity algorithms are: cosine-based similarity, correlation-based similarity and adjusted-cosine similarity [7-9]. In this paper, we briefly mention the inherent drawbacks of the above three similarity algorithms and present an improved similarity algorithm, which can effectively overcome these drawbacks.

The rest of the paper is organized as follows. In section 2, domain ontology building approach is proposed. In section 3, we show the ontology-based user modeling method. Section 4 presents an improved similarity algorithm called Simi-New. Experimental results are provided in section 5. Section 6 states the conclusion of this paper.

II. DOMAIN ONTOLOGY CONSTRUCTION AND DATA PREPROCESSING

In this section we use the OWL Web Ontology Language developed by W3C to build the domain ontology. This language can define the ontology structure, name space, basic elements (classes, individuals, properties) and ontology mapping relationships. We define all kinds of attributes and a variety of property relationships between the ontology concepts. In the paper, we take the rock and mineral fossils domain for example. We make use of automatic construction and hand-built components to build this domain ontology. Firstly, we use the codes of rock and mineral fossils resources to build the meta-data classification hierarchy tree. Secondly, we use the meta-data classification hierarchy tree to build the ontology concept hierarchy tree. At last we add the properties to concept nodes by hand. Figure 1 shows part of the rock and mineral fossils meta-data classification hierarchy tree.
Given the access score vector on leaf nodes \( v' = \{v'_1, v'_2, \ldots, v'_l \} \), we use \( v'_i \) to represent how much the user likes the \( i(1 \leq i \leq t) \)-th concept leaf node, and use the variable \( t \) denotes the number of leaf nodes in ontology concept hierarchy tree. The variable \( v'_i \) can be calculated as:

\[
v'_i = \frac{\log_2(\sum_{R \in \text{Leaf}} F_R)}{\sum_{j=1}^{t-1}(\sum_{R \in \text{Child}} F_R)}
\]  

(1)

Here the variable \( F_R \) means how many times the user visits the resource \( R \) belonging to concept leaf \( l_j \). So far, we have obtained the access score vector on leaf nodes.

As is known to all there are semantic relationships between father-child nodes in ontology concept hierarchy tree, so we can make use of ontology reasoning technology to get the access scores on non-leaf nodes according to the scores on leaf nodes. Given the hierarchy tree has \( t \) leaf nodes, we use \( p_1, p_2, \ldots, p_t \) to define all shortest paths from root to leaf nodes, and use the node set \( \{n_{i_0}, n_{i_1}, \ldots, n_{i_y}\} \) to signify the path \( p_i \) from the root \( n_{i_0} \) to leaf node \( n_{i_y} \) in hierarchy tree. The score of the node \( n_{i_x}(0 \leq x \leq y) \) in the path \( p_i \) is defined as \( s(n_{i_x}) \), which is calculated as following:

\[
s(n_{i_x}) = \begin{cases} 
\alpha * \frac{s(n_{i_{(x+1)}})}{b(n_{i_{(x+1)}})} + 1 & 0 \leq x < y \\
2 & x = y
\end{cases}
\]  

(2)

Here the variable \( n_{i_{(x+1)}} \) denotes the son node of \( n_{i_x} \) in path \( p_i \), \( b(n_{i_{(x+1)}}) \) means the number of \( n_{i_{(x+1)}} \)'s brother in the whole tree, and \( \alpha \) is a reasoning factor which is ascertained in applications (the parameter \( \alpha \) in this paper is equal to 1.8). We can compute for all paths according to the same way. The score of the node \( n_{i_x} \) the user get is given by

\[
s(n_{i_x}) = \sum_{i=1}^{t} s(n_{i_x})
\]  

(3)

After that we can get the access score vector on non-leaf nodes denoted by \( v'' = \{v''_1, v''_2, \ldots, v''_r\} \).

So far, we have obtained score vector \( v' \) on leaf nodes and vector \( v'' \) on non-leaf nodes. After that, we can combine vector \( v' \) with \( v'' \) to generate the ontology-based user modeling.
user model denoted by \( v = \{v'_1, v'_2, \ldots, v'_i, v_v', v''_1, v''_2, \ldots, v''_n\} \), which can also be shown as \( v = \{v_1, v_2, \ldots, v_n\} \).

IV. NEW SIMILARITY ALGORITHM

In section 3 we build the ontology-based user model, in which we use concept nodes to describe the user interests. Generally the domain ontology concept hierarchy tree has hundreds of nodes, but most of users are interested in a few, so the system should be able to handle the problem of the data sparsity. The more sparse the data is, the more difficultly the system finds similar users or similar resources. On the other hand, the different searching hobby also requires a new similarity algorithm. It can be seen from above that the similarity algorithm is one of the key parts in the recommendation systems.

A. Three Commonly Used Similarity Algorithms

There are a number of different ways to compute the similarity between users. Here we present three commonly used methods. These are cosine-based similarity, correlation-based similarity and adjusted-cosine similarity [7-9].

In the cosine-based similarity algorithm, two users are thought of as two vectors in the \( n \) dimensional resource-space. The similarity between them is measured by computing the cosine of the angle between these two vectors. Given \( \vec{i} \) is the score vector rated by user \( i \) in the \( n \) dimensional resource-space, and \( \vec{j} \) is the score vector rated by user \( j \). The similarity between user \( i \) and \( j \) is given by

\[
sim(i, j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}|| ||\vec{j}||}
\]

(4)

Where \( \cdot \) denotes the dot-product of the two vectors.

In the correlation-based similarity algorithm, similarity between two users \( i \) and \( j \) is measured by computing the Pearson – \( r \) correlation \( corr_{ij} \). To make the correlation computation accurate we must first isolate the co-rated cases. Let the set of resources which are both rated by user \( i \) and \( j \) are denoted by \( U_{ij} \) then the correlation similarity is given by

\[
sim(i, j) = \frac{\sum_{u \in U_{ij}} (R_{iu} - \bar{R}_i)(R_{ju} - \bar{R}_j)}{\sqrt{\sum_{u \in U_{ij}} (R_{iu} - \bar{R}_i)^2} \sqrt{\sum_{u \in U_{ij}} (R_{ju} - \bar{R}_j)^2}}
\]

(5)

Here \( R_{iu} \) denotes the rating of \( i \)-th user on resource \( u \), \( \bar{R}_u \) is the average of the \( i \)-th user’s ratings on all resources.

Computing similarity using basic cosine measure has one important drawback—the differences in rating scale between different users are not taken into account. The adjusted cosine similarity offsets this drawback by subtracting the corresponding resource average from each co-rated pair. Formally, the similarity between user \( i \) and \( j \) using this scheme is given by

\[
sim(i, j) = \frac{\sum_{u \in U_{ij}} (R_{iu} - \bar{R}_i)(R_{ju} - \bar{R}_j)}{\sqrt{\sum_{u \in U_{ij}} (R_{iu} - \bar{R}_i)^2} \sqrt{\sum_{u \in U_{ij}} (R_{ju} - \bar{R}_j)^2}}
\]

(6)

Here \( \bar{R}_u \) is the average rating of the resource \( u \).

B. An Improved Similarity Algorithm

This paper presents an improved similarity algorithm called Simi-New. This algorithm is mainly based on the following assumptions [10]:

a) If two users have scored more common resources and less non-common resources, then the similarity between these two users will be higher;

b) If the scores rated by two users on the common resources are closer, then the similarity between these two users will be higher;

c) If the angle between two users’ score vectors is smaller, then the similarity between these two users will be higher;

From the above assumptions, we know we can count the number of resources simultaneously rated by both users \( i \) and \( j \) denoted by \( NOC_{ij} \). We can also count the number of the resources rated by user \( i \) or \( j \) and not simultaneously rated by both the two users, which is denoted by \( NOD_{ij} \).

The ratio of the two numbers, denoted by \( R_{ij} \), is given by

\[
R_{ij} = \frac{NOC_{ij}}{NOD_{ij}}
\]

(7)

The closeness of the scores on the common resources is defined as:

\[
Dis(i, j) = \sqrt{\sum_{u \in U_{ij}} (R_{iu} - R_{ju})^2}
\]

(8)

Here \( U_{ij} \) is the common resources simultaneously rated by both two users \( i \) and \( j \), and \( R_{iu} \) is the score that the user \( u \) rate for the resource \( u \). The angle between two score vectors is counted by Tanimoto coefficient [11], which is given by

\[
S(i, j) = \frac{R_i R_j}{R_i R_i + R_j R_j - R_i R_j}
\]

(9)
Here $\mathbf{R}_i$ is the score vector rated by $i$-th user in the $n$ dimensional resource-space. We define the similarity between users $i$ and $j$ as $\text{sim}(i, j)$ which is computed as following:

$$\text{sim}(i, j) = \frac{w \cdot R_i}{e^{\text{euclid}(i, j)}} + (1 - w)S(i, j)$$

(10)

Here $w(0 < w < 1)$ is a linear weight coefficient, which is specified in the application environment (the variable $w$ in this paper is equal to 0.4).

Simi-New similarity algorithm can offset the drawbacks of the above three commonly used similarity algorithms. Given the score vectors rated by user $x$, $y$ and $z$ are correspondingly $\mathbf{x} = \{0,0,5,0\}$, $\mathbf{y} = \{0,0,1,0\}$ and $\mathbf{z} = \{0,0,4,0\}$. As $\mathbf{x}$ is parallel to $\mathbf{y}$ and $\mathbf{y}$ is parallel to $\mathbf{z}$, so the similarity between $x$ and $y$ and similarity between $y$ and $z$ can’t be distinguished by cosine-based similarity algorithm. However, the similarity between them can be computed by using the Simi-New similarity algorithm. The similarity between the users $x$ and $y$ is $\text{sim}(x, y) = \frac{w}{3} + (1 - w) \frac{5}{21}$, and the similarity between the users $x$ and $z$ is $\text{sim}(x, z) = \frac{w}{3} + (1 - w) \frac{20}{21}$. Obviously, from the above computation we can find out $\text{sim}(x, z) > \text{sim}(x, y)$, which accords with the real condition. Given the score vectors rated by users $x$, $y$ and $z$ are correspondingly $\mathbf{x} = \{0,5,5,0\}$, $\mathbf{y} = \{0,5,1,0\}$ and $\mathbf{z} = \{0,5,4,2\}$. As the vector $\mathbf{x}$ is a self-equal vector, so the similarity between $x$ and $y$ and the similarity between $x$ and $z$ can’t be computed by correlation-based similarity and adjusted-cosine similarity. However, these similarities can be counted and differed by using Simi-New similarity algorithm. The Simi-New similarity algorithm can not only overcome the problem of self-equal vector, but also can adapt to different applications by using adjusted parameter $w$.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In this article all the experimental data used are collected from the rock and mineral fossils resources site (http://www.nimrf.net.cn), and we select five months Web logs from July 1, 2009 to November 31. These logs contain 2695 users who view 75702 pages by 27673 visits.

A. Experiments Of The Similarity Algorithms

In experiment 1, we select 5418 records visited by 335 as the training data, and select 2051 records as the test set to compute Mean Absolute Error (MAE). We choose the User-Based Collaborative Filtering Algorithm as the predicting algorithm. The result of the experiment 1 is showed in Figure 2.

From Figure 2, we can see that the Simi-New algorithm improved by this paper outperforms other three similarity algorithms in the precision of predicting. In experiment 2, we select the same data set as experiment 1 and compare the coverage of four similarity algorithms. The result of the experiment 2 is showed in Figure 3.

From Figure 3, we can see that the Simi-New similarity algorithm is little better than cosine-based similarity algorithm in coverage of recommendation and much better than adjusted-cosine similarity algorithm and correlation-based similarity algorithm. In a word, the Simi-New similarity algorithm is superior to other three commonly used similarity algorithms in the precision and coverage of the recommendations.

B. Experiments Of The Ontology-Based User Model

Experiment 3 and 4 are used to compare the ontology-based user model with the user-resource matrix-based user model to verify the accuracy and validity of the first user model. In experiment 3, we select 5418 records from the first four months Web logs as the training data, and select 2051 records from the last month Web logs as the test set. There...
are 335 users in the experiment data, so the sparsity of the data is 5.05%. We use the above four similarity algorithms to verify the superiority of the ontology-based user model presented in this paper. The result of the experiment 3 is showed in Figure 4.

As shown in the above chart, we can see the ontology-based user model is superior to user-resource matrix-based user model in the condition the sparsity of the data is 5.05%. In experiment 4, we select 7334 records from the first four months Web logs as the training data, and select 2527 records from the last month Web logs as the test set. There are 889 users in the experiment data, so the sparsity of the data is 2.62%. The result of the experiment 4 is showed in Figure 5.

As shown in the above chart, we can see the ontology-based user model is superior to user-resource matrix-based user model in the condition the sparsity of the data is 2.62%. From the above, we can get the conclusion that the ontology-based user model presented in this paper is better than commonly used user-resource matrix-based user model. From the two charts we can also get the conclusion that with the growth of data sparsity the superiority of the Simi-New similarity algorithm is more obvious.

VI. CONCLUSION

This paper presents a new kind of ontology-based user modeling method. We use the ontology concept hierarchy tree to build the user models. This paper newly introduces the ontology and semantic concept to the user modeling method, which makes use of semantic relationship between the concept nodes, so the user model presented in this paper can effectively describe the users’ personalized preferences. We also propose an improved similarity algorithm, which effectively bridges the gap between the traditional similarity algorithms and the precision of personalized recommendation. The new ontology-based user models and improved similarity algorithm effectively improve the quality of the personalized service systems and satisfy the users’ growing personalized needs.

ACKNOWLEDGMENT

The research is supported by the fund of the State Key Laboratory of Software Development Environment SKLSDE-2009ZX-12.

REFERENCES