Guide Me: A Research Work Area Recommender System

Richa Sharma
CSE Department, Chandigarh University, Gharuan, Mohali, 140413, India
E-mail: richas30@yahoo.com

Sharu Vinayak, Rahul Singh
CSE Department, Chandigarh University, Gharuan, Mohali, 140413, India
E-mail: sharuvinayak@gmail.com, rahulsinghcse25@gmail.com

Abstract—With the advent of Industrial Revolution, not only the choices in various fields increased but also the era of computer came into existence thereby revolutionizing the global market. People had numerous choices in front of them that often led to the confusion about what product might actually fulfill their requirements. So the need for having a system which could facilitate the selection criteria and eradicate the dilemma of masses, was realized and ultimately recommender systems of present day world were introduced. So we can refer recommender systems as software tools that narrow down our choices and provide us with the most suitable suggestions as per our requirements. In this paper, we propose a novel recommender system i.e. RWARS (Research Work Area Recommender System) that will recommend research work area to a user based on his/her characteristics similar to those of other users. The characteristics considered here are hobbies, subjects of interests, programming skills and future objectives. The proposed system will use Cosine Similarity approach of Collaborative Filtering.

Index Terms—Collaborative Filtering, Cosine Similarity, Recommender Systems.

I. INTRODUCTION

The term Recommender System is a combination of two words: Recommender and System, where the previous one derived from the Latin word “Recommandare” means-to praise or commend to other as being worthy or desirable, to endorse or to entrust; while the later derived from the Latin word “Systema” means a group of interacting, inter-related or independent elements forming a complex whole. So we can say that recommender systems help making the selection process easier and time saving. The aim of Recommender Systems is to provide the user with the most suitable recommendation, from the plethora of options available.

Recommender systems can be referred to as a system having users, a user interface, a dataset and various recommendation algorithms. It would not be wrong to say that the “Recommender systems are- for the user, to the user and by the user.” It can be elaborated from the fact that the recommendations are made for the users and the dataset used for making the recommendations is also taken from the users themselves in the form of ratings, which can both be implicit or explicit. Then we have the user-interface which acts as the intermediate between the user and the system. The user expresses his preferences, ratings or requirements through the user-interface only and the recommendations are also displayed on the same. Finally, once the user requirements are gathered, some recommendation algorithm is applied. Till date, we have plenty of recommendation algorithms ranging from approach to approach (Table 2). Fig.1 gives an overview of how various aspects of recommender system are related to each other.
implicit or explicit information. The basic idea that lead to the development of Recommender System is that: we often rely on recommendations from our peers before trying out something new e.g.: while buying mobile phones, checking reviews on book my show app before going for movies, even before visiting a doctor and so on. Although there were no recommender systems in old times, still recommendations were there. Kings used to get their recommendations from the ministers e.g. Birbal in case of King Akbar, acted as a recommender system, giving him suggestions regarding the kingdom's policies and important decisions. The most well-known recommender systems of old times were the family relatives suggesting match for marriages. So the concept of recommender systems is not new in real-world scenario, rather it just got automated with advancement in technology and got much better and efficient saving our time.

Researchers started working on Recommender systems as an independent research area during the mid 1970s in Duke University. Since then this area has proven to be the centre of attraction for many researchers and a lot of development has been made in numerous domains using various recommendation approaches developed in the meantime. The first commercial Recommender System was Tapestry, developed in 1992 by Xerox Palo Alto Research Centre [5]. The motivation behind it was the increasing use of e-mail, as a result of which users were being inundated by a huge stream of incoming documents. It used Collaborative filtering.

There are various Recommendation approaches developed so far: Collaborative Filtering, Content-based Filtering, Knowledge-based systems, Hybrid systems, Demographic recommender systems and Community-based systems. Table 1 gives an overview of the most commonly used recommendation approaches.

<table>
<thead>
<tr>
<th>S.no.</th>
<th>Approach</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Collaborative Filtering</td>
<td>i. Provides better results when the no. of users and the no. of ratings are available in abundance.</td>
<td>i. Cold start problem both in case of new users and new items.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. No knowledge engineering required</td>
<td>ii. Sparsity problems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i. This is basically people-to-people correlation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. It uses the wisdom of the crowd to recommend items to the user.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii. The ratings can either be explicit or implicit.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>iv. It assumes that people who had similar tastes in past, will also have similar tastes in future also.</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Content-based Filtering</td>
<td>i. It does not rely on other users' preferences.</td>
<td>i. Cold start for new users.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. Comparison between items possible</td>
<td>ii. No surprises.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>i. It is based on the concept &quot;Show me more of what I have liked&quot;.</td>
<td>iii. Grey-sheep problem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. It takes into account user preferences and on the basis of that, makes the recommendations.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii. It recommends those items which are similar to the user preferences based on their past behavior.</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Knowledge based</td>
<td>i. Deterministic recommendations</td>
<td>i. Cost of knowledge acquisition from domain experts, users etc. are too high.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. Assured quality</td>
<td>ii. Basically static</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii. No cold start</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>i. It is based on the concept &quot;Tell me what fits my needs&quot;.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. No user profile is maintained, rather user interaction is there.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii. It uses knowledge of users &amp; items.</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Hybrid Recommender System</td>
<td>i. Avoids most of the shortcomings of other approaches.</td>
<td>i. A little complex to apply on the same dataset.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii. A hybrid system combining techniques A and B tries to use the advantages of A to fix the disadvantages of B.</td>
<td></td>
</tr>
</tbody>
</table>
Table 2. Recommendation Algorithms

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Algorithms</th>
</tr>
</thead>
</table>
| 1.    | Pearson co-relation is used to find similarity between users based on their ratings.  
\[
sim(a,b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}
\]  
Here,  
- \( a,b \) - users  
- \( r_{a,p} \) - rating of user a for product p  
- \( \bar{r}_a,\bar{r}_b \) - average user rating  
- \( P \) - set of products rated by both a and b |
| 2.    | Constrained Pearson co-relation is also used to find similarity between users, when co-relation between them increases.  
\[
sim(a,b) = \frac{\sum_{p \in P} (r_{a,p}-mp)(r_{b,p}-mp)}{\sqrt{\sum_{p \in P} (r_{a,p}-mp)^2} \sqrt{\sum_{p \in P} (r_{b,p}-mp)^2}}
\]  
Here,  
- \( mp \) - midpoint of rating scale |
| 3.    | Cosine similarity is a vector based approach based on linear algebra.  
\[
sim(a,b) = \frac{r_a.r_b}{r_a^2 r_b^2}
\] |
| 4.    | Spearman Co-relation is a non-parametric method that computes a measure of co-relation between ranks.  
\[
sim(a,b) = \frac{\sum (rank_{a_i}-rank_{b_i})(rank_{a_j}-rank_{b_j})}{\sqrt{\sum (rank_{a_i}-rank_{a_j})^2} \sqrt{\sum (rank_{b_i}-rank_{b_j})^2}}
\] |
| 5.    | K Nearest neighbour approach is a simple algorithm used as a non-parametric technique to calculate the similarity measure or distance between the neighbors (users, items or products). Nearest neighbor can be found out by calculating the distance between all the neighbors by using:  
- Euclidean distance  
\[
\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}
\]  
- Manhattan distance  
\[
\sum_{i=1}^{k} |x_i - y_i|
\] |

From social networking sites to marriage portals, from e-commerce sites to movie, music or book recommender system, we have recommendations for almost every field. They have a really crucial role in internet sites like Amazon, Pandora, LinkedIn, Netflix, Matrimonial sites, CiteSeer, Social networking sites, YouTube, Yahoo, Jester and a lot more. These websites use either one of the recommendation approach or a combination of two or more. For example, Amazon uses Item to Item collaborative filtering and content-based filtering, CiteSeer uses collaborative and content-based filtering, Pandora uses content-based filtering, Youtube uses personalized recommendations while both Jester and LinkedIn use collaborative filtering. Recommender Systems have evolved so much over the past two decades and now have included Artificial intelligence, Information retrieval and Human-computer interaction in their journey [1], and hence have got more efficient and have gained more popularity.

Table 3. Some common Recommender Systems developed so far

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Recommender system</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Facebook</td>
<td>Collaborative filtering, Content based</td>
</tr>
<tr>
<td>2.</td>
<td>Twitter</td>
<td>Collaborative filtering</td>
</tr>
<tr>
<td>3.</td>
<td>LinkedIn</td>
<td>Collaborative filtering</td>
</tr>
<tr>
<td>4.</td>
<td>Netflix</td>
<td>Hybrid recommender system</td>
</tr>
<tr>
<td>5.</td>
<td>MovieLens</td>
<td>Collaborative filtering</td>
</tr>
<tr>
<td>6.</td>
<td>Pandora</td>
<td>Content-based recommendation</td>
</tr>
<tr>
<td>7.</td>
<td>Last.fm</td>
<td>Collaborative filtering</td>
</tr>
<tr>
<td>8.</td>
<td>Amazon</td>
<td>Item to item collaborative filtering, Content-based recommendation</td>
</tr>
<tr>
<td>9.</td>
<td>eBay</td>
<td>Collaborative filtering, Demographic recommendation</td>
</tr>
<tr>
<td>10.</td>
<td>CiteSeer</td>
<td>Content-based, Collaborative filtering</td>
</tr>
<tr>
<td>11.</td>
<td>Docear</td>
<td>Content-based</td>
</tr>
<tr>
<td>12.</td>
<td>Techlens</td>
<td>Content-based, Collaborative filtering</td>
</tr>
<tr>
<td>13.</td>
<td>Youtube</td>
<td>Personalized recommendation</td>
</tr>
<tr>
<td>14.</td>
<td>Jester</td>
<td>Collaborative filtering</td>
</tr>
</tbody>
</table>
Since, Recommender Systems have reduced our effort of going through each and every item and selecting the most suitable one, it would be great if there is a Recommender System that can suggest an area to a student that he or she can do his research in. In this paper, we propose RWARS (Research Work Area Recommender System) that recommends an area for research work by using cosine-similarity. This system would find the similarity between the users in terms of their future objectives, the type of research they want to do, their hobbies, subjects of interests and their programming skills. As the dataset, we have collected our own data by conducting a small survey of Post-graduate students who have already worked in some research area. We anticipate that our proposed system will prove to be beneficial.

This paper is organized as follows. Section II evaluates the various works related to the proposed system. Section III reports the framework of the proposed system. In section IV, various aspects of the system are discussed and the processes in each phase are elaborated. Finally, the paper is concluded in Section V.

II. RELATED WORK

In this section, we briefly describe some of the work related to recommender systems in general, recommender systems related to the field of education and recommender systems using collaborative filtering, keyword extraction and works that have deployed cosine similarity.

Recommender system is one of the most popular and mighty tool, that has taken over e-commerce [1]. The basic idea that led to its development is to help the users select the most suitable product or item from a plethora of options in front of them [1] [2]. The most popular approaches of Recommender Systems are: Collaborative Filtering, Content-based systems, Hybrid systems and Knowledge-based systems.

Collaborative filtering is also known as ‘people-to-people co-relation’ [1]. It is based on a simple idea that users having identical interests in the past, tend to have alike interests in the future also [1] [2] [3] [4]. Content-based systems follow the concept “Show me more of what I have liked” [4]. The basic idea is to recommend products to a user, which are identical to the ones that user has already liked in the past [1] [2]. The similarity between two or more items can be calculated using the features they have in common.

Knowledge-based systems follow the concept “Tell me what fits my needs” [4]. Such systems use domain knowledge of a specific product to make recommendations to the user [1]. The system gets the user requirements, matches it with its knowledge base and makes the recommendations of the most appropriate and suitable product to the user, keeping user preferences in consideration. Hybrid system gives the best of both worlds. As the name suggests, these systems combine the features of two recommendation approaches. The basic purpose of these systems is to combine the features of two systems (recommendation approaches) in such a way that the shortcomings of one are overcome by the other [1] [2] [4].

Collaborative filtering suffers from cold-start problem while there is no element of surprise in case of content-based systems [4]. So combining the features of these two approaches often prove to be more efficient and useful. Using the features of explicit ratings and implicit ratings of collaborative filtering and content based systems respectively, the problem of data-sparsity and cold-start problem can be tackled. This approach was used by Mojtaba et al. [6] in their work where, they presented a recommender system for recommending learning material to the users. Their work deployed genetic algorithm along with a multi-dimensional information model. Joonskeok et al. [7] proposed a Personalized Academic Research Paper Recommender system that used collaborative filtering. The system used web-crawler to retrieve various research papers from the internet, and then found the similarity between two research papers using Nearest neighbor approach and finally made the recommendations. Similar work was forth put by Nitin et al. [8] who too worked for reducing the sparsity of the data. For that, they introduced Subspace Clustering Algorithm (SCuBA) for recommending research papers to the user.

Nazpar Yazdanfar et al. [9] proposed a Link Recommender system that recommends URLs to twitter users using nearest neighborhood approach. They considered hashtags as the representatives of the URLs and used Euclidean, Jaccard, Dice coefficient and Cosine similarity measures to find the item-hashtag and item-user similarity. Pu Wang [10] presented a personalized collaborative approach that was based on the clustering of the customers to overcome the scalability issue. Approaches like K-means clustering, Cosine and Co-relation similarity and Nearest neighbor approach were deployed and along with scalability, issues like data sparsity and cold-start problem were also resolved.

Kwanghee et al. [11] proposed a personalized research paper recommender system that recommends research paper to the user based on his profile. The system used the concept of keyword extraction to find the similarity between the topics that the user has desire to work in and the research papers available on the web. To find the similarity, Cosine similarity was used and this system proved to be quite efficient in making the search for research papers easier.

One of the newest works being done in Recommender systems is that of cross-domain recommendation i.e. collaborating two different domains together and making the recommendations based on the relation between the two. Hla Hla Moe et al.[12] proposed an approach for building ontologies using case-based approach for cross-domain recommendation for cosmetics related to various skin problems. For this work, Ford-Fulkerson algorithm has been used and the results showed that the system proved to be quite efficient and gave better recommendations as compared to other cross-domain systems. Naziza Abderrahim et al. [13] proposed a social-trust aware system that based on the social qualities of web services, recommended web services to
the user taking their overall experience into consideration.

Fatiha Bousbahi et al. [14] proposed a recommender system for online courses to the user based on his requests. The proposed system used case-based approach for making the recommendations to the user as per his user profile, knowledge and requirements. Haifeng Liu et al. [15] presented a novel model for calculating the new user similarity. The work focused primarily on improving the quality of recommendations in case of insufficient user-ratings. Diksha Nagpal et al. [16] proposed a faculty recommender system. This system used both user-user collaborative filtering and item-item collaborative filtering. Results of the study revealed that item-item collaborative filtering performed better than user-user collaborative filtering.

Based on the previous work done, we concluded that some more work can still be done in the field of education by developing a Research Work Area Recommender System. We have article recommender system, recommender system for digital libraries and even research paper recommender systems. But selecting what research area to work on is in itself a very tedious task as it requires going through a lot of research works to decide which one interests you the most. With the proposed system, we can not only save time and effort, but we can also make this selection process a lot easier and simpler. The user has to simply answer a short questionnaire and in no time, he or she would be provided with the most suitable recommendations. So we propose RWARS (Research Work Area Recommender System) that would recommend research work areas to the users based on their interests, hobbies, programming skills, subjects of interests and future objectives.

III. SYSTEM ARCHITECTURE

RWARS (Research Work Area Recommender System) is composed of a User Interface, Extractor, Filter, Profile Manager and DB Manager.

Figure 1 gives an overview of the architecture of the proposed system.

The User Interface interacts with the users, takes the input information and sends it to the Extractor that extracts the words or the terms. The basic purpose is to first extract the words from the form filled by the new user and then extract similar words from the dataset. We have designed a very simple user-interface using ASP.NET. The User Interface is composed of a simple form asking the user questions similar to the ones that were asked in the survey. All the user has to do is answer the questions asked in the form and then click on the submit button. Once that is done, the extractor would come into action and extract all the words i.e. whatever his/her responses to the questions are. So the hobbies, subjects of interests, future objectives and programming skills as selected by the user, would be the words or terms to be extracted.

Filter undergoes filtering process to make sure that only the relevant data are taken. Relevant data in our case is only the data that are to be compared. For example, we are only considering the hobbies, subjects of interests, skills and the future objectives. Parameters like name or contact information are not being taken into account.

User Profile stores the information of the users along with the recommendations made. For each user, who would use the system, the results would be saved along with the other data (as filled by him/her in the form). This way a user can keep track of recommendations made to him, based how he filled the form. Also the user can see how his varying interests in subjects or how his programming skills affect the recommendations made. It does not at all depend on an individual that what recommendations would be made, rather his characteristics and skills would be the deciding factor. For example, consider a user “x”. Now the first time he uses the system, he is recommended Data Mining, suppose. But the next time he is recommended Recommender Systems. Now it does not imply that the system does not work effectively, rather it means that the second time when the user filled the data, he might have learned some new programming skill or he might have developed his interest in a new subject or he might have changed his objective. So for each user or a user with varying interests, the results would vary.

The DB Manager manages the dataset that we have collected after conducting the survey of students of Masters Degree. The survey had simple questions asking the students about their Research area, Future objectives (job or PhD), Hobbies, Programming skills, what kind of research they wanted to do (Academic or Industrial) and if they would simply opt for the research area suggested by the teacher they wish to work with.
IV. RESEARCH WORK AREA RECOMMENDATION

The proposed recommender system will use Case-based approach to find the similarity between a user who wishes to get the recommendations and the ones who have already opted for some research work area. As shown in Fig.3, RWARS is divided into three phases: Data gathering, Fetching new user data and Making recommendations.

A. Data gathering

For the database of the proposed system, we have first prepared a small survey and then asked the Post-graduate students to fill the survey. The data gathered from the survey is entered into the system manually and will be used as the dataset. Fig.4 and Fig.5 shows the questions that were asked in the survey while in Fig.6, we have a .CSV file of the data that have been gathered.

B. Fetching new user data

We have designed a simple User-interface, where a new user would fill the form as shown in Fig.7. Once the user fills the form and clicks on the submit button, his information would be stored in a temporary database. All the algorithms would work at the back-end and once this is done, the results would be displayed on the user-interface only.

C. Making Recommendations

As shown in Fig.2, for making the recommendations first theExtractor would extract all the words from the Temporary dataset. For convenience, we would assign rank 1 to each word. Once this is done, we would find out the similarity between the new user and the ones in the
dataset using Cosine similarity. It is a vector based approach based on linear algebra. Users are considered as vectors and similarity between them is measured by the cosine distance or dot product of the angles between the rating vectors \[2\] \[3\]. It is given by:
\[
sim (a, b) = \frac{r_a \cdot r_b}{|r_a||r_b|}
\]
Equation 1,
\(r_a\) is the rating given by user a
\(r_b\) is the rating given by user b

In the proposed work, we would consider the new user as user a and rest all the users of the dataset would be considered as user b. Then we would apply the cosine similarity algorithm to find the similarity between the users (a and b). Finally, for a user a having maximum similarity with a user b, recommendation for the research work area would be made.

V. CONCLUSION
In this paper, we proposed a novel recommender system RWARS that will recommend a research work area to the user based on the similarity of user characteristics between the new user and the ones who have already worked in some research area. A user has to answer a simple questionnaire asking him about his skills, subjects of interests, future objectives etc. and based upon these answers he will be provided with the recommendations.

The proposed system is still in its development stage and a lot of work still needs to be done. We have conducted a survey of Masters Students and used that as the dataset. Also, at present we have planned to use Cosine-similarity approach only to make the recommendations. But in future we aim to apply other recommendation approaches too and then compare the results obtained from each approach. The parameters that would be considered while deciding which approach is better than the other are coverage, precision, recall and effectiveness. Now just the name of the research area would not be sufficient for the user to take any decision, so in future, we have planned to give the links of a literature review of the research areas too along with the names thereby helping the user to the maximum.

REFERENCES

Authors’ Profiles
Richa Sharma: Belongs to Bilaspur, Himachal Pradesh, India and born on March 6, 1992. She is a Research student for Masters Degree for Computer Science Engineering in Chandigarh University. She received her Bachelor’s Degree from Punjab Technical University in 2014. The field of her thesis is Recommender Systems. Her areas of interests for research are R language for
building Recommender Systems, Data Privacy and Security.

Sharu Vinayak: Belongs to Ludhiana, Punjab, India and born on October 8, 1992. She is currently pursuing Masters in Computer Science and Engineering from Chandigarh University and has done Bachelor’s degree in Information Technology from Ludhiana College of Engineering and Technology. Programming and learning about new technologies interests the author.

Rahul Singh: Belongs to Kanpur, Uttar Pradesh, India and born on April 15, 1989. He is an Assistant Professor in CSE Department of Chandigarh University. He has received his Masters degree from Thapar University. His areas of interests are Theory of Computation and Software Engineering.

How to cite this paper: Richa Sharma, Sharu Vinayak, Rahul Singh, "Guide Me: A Research Work Area Recommender System", International Journal of Intelligent Systems and Applications (IJISA), Vol.8, No.9, pp.30-37, 2016. DOI: 10.5815/ijisa.2016.09.04