Algorithm and Optimization for Big Data (AOBD) Final Paper

Roll No:- 1401098
Name:- Pinak Divecha
School Of Engineering and Applied Science
Ahmedabad University

***Abstract*—Today, there is a big variety of different approaches and algorithms of data filtering and recommendations giving. In this paper, recommendation systems are defined as the techniques used to predict the required skill of an individual. This document discusses collaborative filtering based recommendation systems, i.e., systems which takes into account user's past behavior to make choices. Here, to solved module-I i have used collaborative approach and to solve module-II i have used the fuzzy logic as my first approach and second approach is collaborative-based approach.**

***Index Terms*—Recommendation System, Fuzzy Logic, String Matching, Collaborative Filtering, Linear Regression, regularization, Mean Normalization, Gradient Decent.**

I. Introduction

Recommendation systems assist and augment this natural social process to help people sift through available books, articles, web pages, movies, music, restaurants, jokes, grocery products, skill required, and so forth to find the most interesting and valuable information for them. Recommendation systems identify recommendations autonomously for individual users based on past purchases and searches, and on other users' behavior.

II. TRADITIONAL RECOMMENDER APPROACHES

Typically, recommender systems are classified according to the technique used to create a recommendation: Content-based system, Collaborative-based system and Hybrid-based system.

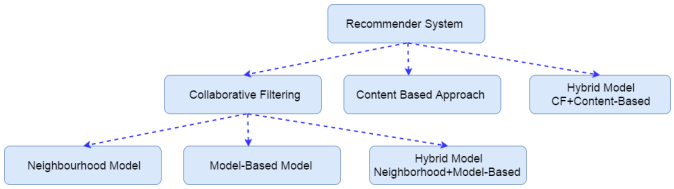

Fig. 1: Basic approaches for recommendation system

1) **Collaborative-based filtering:** Collaborative filtering arrives at a recommendation that's based on a model of prior user behavior. The model can be constructed solely from a single user's behavior or more effectively also from the behavior of other users who have similar traits. When it takes other users' behavior into account, collaborative filtering uses group knowledge to form a recommendation based on like users.
2) **Content-based filtering:** Content-based filtering constructs a recommendation on the basis of a user's behavior. For example, this approach might use historical browsing information, such as which blogs the user reads and the characteristics of those blogs. If a user commonly reads articles about Linux or is likely to leave comments on blogs about software engineering, content-based filtering can use this history to identify and recommend similar content.

3) **Hybrid-based filtering:** Hybrid approaches that combine collaborative and content-based filtering are also increasing the efficiency (and complexity) of recommender systems.Incorporating the results of collaborative and content-based filtering creates the potential for a more accurate recommendation.

III. Data Set

Here, we were given data about 39 different job posts like Automation Test Engineer, Java Developer, Front End Developer, System Analyst and etc. in JSon file format. Each of these files had information about each and every candidate. Each file contains information about candidates id, current skills, work-experience, Education etc. Then the data was converted into csv file format for cleaning up. Then we implemented python code to clean up and extract the required fields so that it would be easy for us to implement it on our approach. And the csv file that we obtained contains 3 columns:

- Candidates Id
- Skills
- Year

IV. Module-I

Here, I have used collaborative learning algorithm to find the skill of a particular candidate. Initially,I collect the skill of the users and store in the 2D matrix. In the given matrix row are considered as skills and column are considered as users. After getting the matrix, it is converted into the binary matrix(1 or 0). In this matrix if the users has a particular skill

then its value will be 1 else 0. This type of matrix is quite sparse, since not all users have all the skills. After getting the 2D dataset matrix i.e. skill x users. Now we implement collaborative filtering based recommendation system model. For this model, we need to compute the top recommended skills. In the following part, we will explicitly show how we build our recommendation system, and compare different models in the following subsection.

In my code, i have taken 10 users and 10 skills which are required for that particular job. So initially our data matrix Y is in the form of sparse matrix, where rows are taken as skills and columns are taken as users. Matrix R is contain the binary values i.e. either one or zero. Then we take the feature vector of experience. In that we map X matrix as average experience required of that particular job and Theta vector as experience of each user. Now we have to predict the skill whether it is required or not for a particular user in that matrix. So now it become a classification problem for predicting future skills preference. And to solve this problem we have to use linear regression method. We now we perform the linear regression on matrix X and theta. As, linear regression focuses on the minimize this cost function and minimize the sum of squared errors.

After multiple iterations of gradient descent, we would have found the values of matrices user and experience that minimize our cost function. Essentially, we will have learned the appropriate values of user and experience to make accurate predictions on the skills for every user. Now, lets use our learning algorithm learn to predict top skills. We just have enter the candidate ID and our predictor will predict the skills.

The notations i have used in the code are as follows:

- R(i, j): User j has achieved ith skill.
- Y(i,j) : experience by user j in skill i.
- The value of R(i,j) is 1 if and only if that particular individual has acquired that skill. And Y(i,j) is defined iff R(I,j) is 1.

A. *Algorithm*

1: **initialize**: (i, j): User j has achieved ith skill.
2: Y(i,j) : experience by user j in skill i.
3: if particular person has acquired that skill
4: The value of R(i,j) is 1.
5: else
6: The value of R(i,j) is 1.
7: And Y(i,j) is defined iff R(I,j) is 1.
8: **output:** Skills

B. *Results*

V. MODULE-II

A. ***Approach-I***

In second module, when user enters a career goal and based on that career goal the platform suggests a career path.

```
Top recommendations for you:
Predicting rating 8.5 for movie Python
Predicting rating 5.9 for movie ESD
Predicting rating 4.9 for movie Sql
Predicting rating 4.7 for movie Oracle
Predicting rating 3.5 for movie C++
Predicting rating 3.5 for movie Java
Predicting rating 3.2 for movie C
Predicting rating 1.7 for movie Db2
Predicting rating 1.2 for movie Hadoop
Predicting rating -2.2 for movie Shell


Original ratings provided:
Rated 5 for Sql
Rated 2 for Db2
Rated 3 for Oracle
```

Fig. 2: Output for Module-I

```
Y =

     0     3     1     1     5     1     0     3     1     5
     4     3     2     2     5     4     3     2     1     5
     0     5     4     0     0     3     5     1     3     4
     0     3     4     5     4     3     4     2     1     1
     3     4     0     1     1     4     1     2     4     3
     0     2     2     0     2     3     2     0     5     0
     4     2     3     2     3     1     2     3     4     2
     4     4     2     1     5     0     5     1     2     1
     4     0     3     2     2     5     0     2     3     0
     0     0     3     2     5     1     5     3     0     1

     0     1     1     1     1     1     0     1     1     1
     1     1     1     1     1     1     1     1     1     1
     0     1     1     0     0     1     1     1     1     1
     0     1     1     1     1     1     1     1     1     1
     1     1     0     1     1     1     1     1     1     1
     0     1     1     0     1     1     1     0     1     0
     1     1     1     1     1     1     1     1     1     1
     1     1     1     1     1     0     1     1     1     1
     1     0     1     1     1     1     0     1     1     0
     0     0     1     1     1     1     1     1     0     1
```

Fig. 3: Output for Module-I

To implement this i have used Mean, fuzzy logic and string matching. In this recommendation system those skills are suggested first which have been been adopted by maximum no of users. I have also used auto-complete recommendation, in order to help the user while entering the career goal. For this module i have clean up my data-set according to my second module requirements. So when user write software in search bar, system suggests career goals having Software keywords. Example Software Developer, Software Engineer, Software Architect

Algorithm:-

The Steps in Algorithm are:

1) Initially parse the JSON.
2) Then i have map the array of skills corresponding to career goal.
3) Sorted the skills on the basis of number of times they occur in give role and put them accordingly in order.
4) At the time of search, we have auto-complete of the career goal, that will have the user a lot at the time of typing.

Here career goal contains all the element in the form of javaScript object. ForEach function iterate through every item of object which contains Role (Job Position) and Skills. Role (Job Position) is taken here as a career goal.

The role is pushed into array for searching purpose and checking whether the same role is already present in the map. If yes, then append the skill set with given skill and concat in previous skills of the same role. If this is for the first time, simply add the skill corresponding the new role.

Next we remove the duplicate value out of the array of job roles. Thus we have unique career goals. We are doing search route in order to set all the roles for auto-complete. The required skills set according to the job type will be returned by fetching the job string from the route, Here the reduce function is going to map the skills set according to the number of times the given skill occurs. for example, if someone wants to become Software Developer and in that role JAVA comes 5 times and SQL comes 3 times then first it will suggest JAVA.

Results:

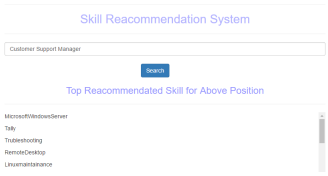

Fig. 4: Output for Module-II

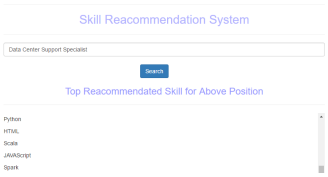

Fig. 5: Output for Module-II

B. Approach-II

Our module-II is online version of module-I because he/she want to aquire skill for the particular job position even if the new user have or have not the required skill. This module also have the same step as above module where firstly i have a sparse matrix which column contains the number of users and row has number of skills.Now new user has come we assign the required parameter like X, Theta and Y. Then, we calculate linear regression followed by gradient decent to minimize our cost function.

Now, lets use our learning algorithm learn to predict top skills for this new job position. We just have enter the new candidate ID and our predictor will predict the skills based on our learning algorithm.

C. Algorithm

1: For every skill i that user u has no preference or some preference
2: For every other user y that has a preference for i
3: Compute a similarity s between u and y
4: Add v's preference for i, weighted by s, to a running average
5: Return the top skills, ranked by weighted average

VI. CHALLENGES OF COLLABORATIVE FILTERING

1) **Data sparsity:-**Recommendation system are based on the large dataset and because of that user-item matrix used for collaborative filtering will be large and sparse which is bring the challenges in the performances of the recommendation.

2) **"Cold start" problem:-** As Collaborative-filtering is a powerful way of recommending items based on user skill the user has, but sometimes there is no skill and this is called the cold start problem, and it can apply both to new user and to skill..

VII. OTHER APPROACHES FOR OUR PROBLEM

You can implement a clustering algorithm such as k-means or DBSCAN to group users with similar features together, and thereby recommend the same skills to users belonging to the same cluster.

VIII. OTHER APPLICATION OF RECOMMENDATION SYSTEM

There are several others important applications of recommendation systems which are :

1) **Product Recommendations:** Perhaps the most important use of recommendation systems is at on-line retailers.We have noted how Amazon or similar on-line vendors strive to present each returning user with some suggestions of products that they might like to buy.

2) **News Articles:** News services have attempted to identify articles of interest to readers, based on the articles that they have read in the past. The similarity might be based on the similarity of important words in the documents, or on the articles that are read by people with similar reading tastes. The same principles apply to recommending blogs from among the millions of blogs available, videos on YouTube, or other sites where content is provided regularly

REFERENCES

[1] https://en.wikipedia.org/wiki/Collaborativefiltering.
[2] https://www.ibm.com/developerworks/library/os-recommender1.