Algorithms and Optimization for Big Data (AOBD) Final Paper

School Of Engineering And Applied Science
Ahmedabad University
Vatsal Patel-1401123

***Abstract*—In today's generation technology is getting improved every single day and with latest technology everything became easier. Now What if you can find your desired job and career path easily with multiple option? In this paper i have introduced a approach of problem of job recommendation where in the first module considering the career path, some particular skillset are suggested to the user and in second module from given career goal and addition information of the user, algorithm suggest what career path user should choose in order to reach that particular goal.**

***Index Terms*—Collaborative Filtering, content based approach, linear regression , gradient descent.**

I. Introduction

The motivation to create job recommender system is to create module for linkdin where there are around 135 million user and to connect the worlds professionals to make them more productive and successful. Nowadays we are daily overwhelmed by huge amounts of information which we are usually unable to process. This fact makes it difficult to identify those information pieces that are really useful, interesting and meaningful to us. Such information pieces may include books, newspapers, web sites, songs, movies, or the CVs of candidates for a job position. A recommender system is based on scientific domain such as AI, data mining, pattern recognition and machine learning.

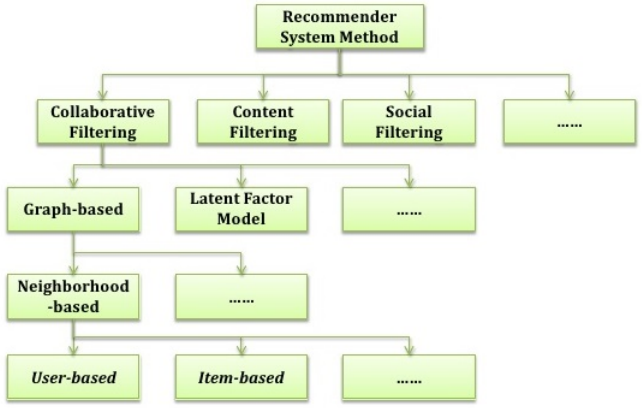
II. Recommender system

A recommender system must be personalized and be able to help user select among discrete option, in other word recommender system recommends you relevant information based on various activities related to that application.

A. Application of Recommendation System

- Now a days recommendation systems is mostly used by online retailers such as Amazon, flipkart, Linkdin and many more uses this for suggestions of products that they might like to buy. These suggestions are not random, but are based on the purchasing decisions made by similar customers.
- Netflix offers its customers recommendations of movies they might like. These recommendations are based on ratings provided by users.

Some of the method for recommender system are,
1] Collaborative Filtering
2] Content based approach
3] Knowledge based recommender
4] Hybrid recommender.



B. Collaborative Filtering

collaborative filtering is a method of making automatic predictions (filtering) about the interests of a user by collecting preferences or taste information from many users (collaborating). It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. In our case a person who wants a job might look for other person with same skillset and his/her job. This information is used in the decision on which kind of job he can be done by him. In the more general sense, collaborative filtering is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources, etc .

C. Content based filtering

Content-based filtering method is basically based on description of the item and users preference. In content-based recommender system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past.

D. Hybrid recommender

Hybrid recommender is basically combination of collaborative and content-based filtering. In this recommender system efficiency and complexity are increased in compare to other

system. The hybrid approach could also be used to address collaborative filtering that starts with sparse data known as cold start by enabling the results to be weighted initially toward content-based filtering, then shifting the weight toward collaborative filtering as the available user data set matures.

III. Data set

The data set used to implement this approach contains different job position. In this there is different type of information of many user such as candidateID , skill etc. Further, this files are given in Json format and this file are first converted into csv file. Then i have implemented a python code to extract only required element from user profile. Also, the provided data required to be cleaned. To implement both of the module given elements are extracted,

1] CandidateID
2] Skills
3] Experience year

IV. Module 1

In this module, system basically reads user profile and according to the job he/she wants system will suggest the necessary skillset which a user needs to acquire.

So in order to solve this problem i have approached collaborative algorithm. So the first step is to fetch the required data element of the user in csv format using python code. Further the data is stored in two 2D matrices in which column represent different skills and row represent candidateID. If the user have a particular skill then that element or cell will be given value of 1 and 0 otherwise. After getting the 2D dataset matrix (skills X users), i implemented Collaborative Filtering based recommendation system model. In this model the we count that which skill is acquired by most number of people and that will be top recomended skill.

So first of all we import the data set and the parameter and that data is in form of matrix which has skills as rows and users as the columns, So initial our data matrix Y exists in the form of a sparse matrix, where rows correspond to skills, columns correspond to user and the matrix entries are either zeros or ones. Then we take experience as our feature vector. In that we map X matrix as average experience required of that particular job and Theta vector as experience of each user. Further we will see that what skill are required for particular career path. So this is actually become classification problem for predicting future skill preference, to solve a problem we have used linear regression method. We got X matrix Theta matrix we perform linear regression. Linear regression is used to minimize cost function and minimize the sum of squared errors. After multiple iterations of gradient descent, we will further find the values of matrices user that minimize our cost function. so we learned the appropriate values of user and make accurate predictions on the skills for every user.

V. Module 2

In this module, a career paths recommended to user based on career goal and other information about the user in other word that what kind of job user should do in order to reach his career goal. This includes his current skills as well as current job profile.

So to suggest career path we first of we have to fetch the same data as module 1 and when new user arrives we assign the required parameter like X, Theta and Y. Then, we calculate linear regression followed by gradient decent to minimize our cost function.

Another approach to solve this problem:
In this algorithm we use concept f Mean, fuzzy logic and string matching. In order to help the user while entering the career goal, i have also used auto complete recommendation. So when user write Developer in search bar, system suggests career goals having developer keywords in it. Eg Software Developer, GUI Developer, Java Developer

Here career goal contains all the element in the form of javaScript object. forEach function iterate through every Job Position and Skills. 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 connect in previous skills of the same role. if this is for the first time, simply add the skill corresponding the new role.

Further we remove the duplicate in order to get 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.

VI. Algorithm

A. Module-1

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. Module-2

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

C. Module-2 Approach-2

1: Initially parse the JSON
2: Using forEach loop map the array of skills corresponding to career goal.
3: Internally sort 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 autocomplete of the career goal, that will have the user alot at the time of typing.

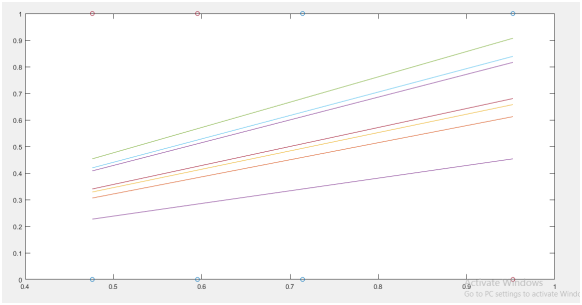
VII. Result

A. Module-1

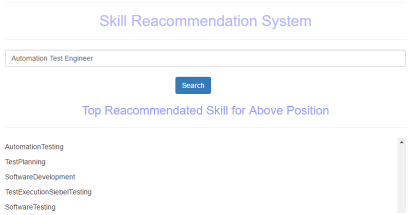
```
Top recommendations for you:
Predicting rating 6.4 for skills Shell
Predicting rating 4.9 for skills Db2
Predicting rating 4.6 for skills Hadoop
Predicting rating 4.1 for skills ESD
Predicting rating 3.8 for skills Sql
Predicting rating 3.4 for skills Python
Predicting rating 2.3 for skills C
Predicting rating 1.6 for skills Java
Predicting rating -0.1 for skills Oracle
Predicting rating -1.8 for skills C++


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

B. Module-2



C. Module-2 Approach-2



D. Challenges

The recommender system is working with large datasets thats why the user-item matrix used for collaborative filtering could be extremely large and sparse, which brings about the challenges in the performances of the recommendation. Sometime this algorithm may fail to recommend due to lack of data availability. As the numbers of users and items grow, traditional Collaborative filter algorithms will suffer serious scalability problems.

References

[1] Xiaoyuan Su, Taghi M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in Artificial Intelligence archive, 2009.
[2] https://en.wikipedia.org/wiki/Collaborativefiltering.
[3] http://infolab.stanford.edu/ ullman/mmds/ch9.pdf