

Algorithms and Optimization for Big Data (AOBD)

Final Paper

Kirtan H. Modi
1401122

School of Engineering And Applied Sciences
Ahmedabad University

Abstract—Today whenever we start youtube it suggest us that "Videos recommended for you" or even if we start Netflix we see "Movies recommended for you". So,when this kind of things are available in entertainment area I think this kind of things should be available in educational or job related areas. So, here we are trying to develop and implement the algorithm to recommend skills to a user. In first module we will suggest a user some skills he/she needs to gain using collaborative filtering approach. In the second module we use the same approach, but here we ask the user to enter his/her goal and on the basis of goal we will recommend skills to gain.

Keywords: Linear Regression, Gradient Descent, Recommendation System, Collaborative Filtering, Content-based Filtering, Regularization, Mean Normalization.

I. INTRODUCTION

Recommendation systems are one of the most common and easily understandable applications of big data. Every year thousands of engineers, doctors, CAs pass-out with decent grades. But ultimately what happens they have to roam around cities to cities to find a job. Sometimes what happens a person may have certain amount of skills but he/she doesnot know the future path. So, our approach will help them to choose a future career path on the basis of their current skills and will even recommend skills to hands on.

There are different kinds of applications of Recommendation Systems in real-life. Which are like:

- Amazon: "Customers Who Bought This Item Also Bought"

Amazons algorithm crunches data on all of its millions of customer baskets, to figure out which items are frequently bought together. This can lead to huge returns- for example, if youre buying an electrical item, and see a recommendation for the cables or batteries it requires beneath it, youre very likely to purchase both the core product and the accessories from Amazon.

- LinkedIn: "Jobs You May be Interested In"
he Jobs You May Be Interested In feature shows jobs posted on LinkedIn that match your profile in some way. These recommendations shown based on the titles and descriptions in your previous experience, and the skills other users have endorsed.

Here,our algorithm is able to suggest skills which is similar to users skill. It's an online version. The rest of the paper is organized as follows. Section II Science behind recommendations. Section III Dataset. Section IV Module 1. Section V

Module 2. Section VI Algorithm. Section VII Main Results. Section VIII Challenges.

II. SCIENCE BEHIND RECOMMENDATIONS:

This is the general scenario of any recommendation system. There are two major ways most of recommendation engines work. They can either rely on the properties of the items that each user likes, discovering what else the user may like (content-based filtering method) or , recommendation engines can rely on likes and desires of other users in order to compute a similarity index between users and recommend items to them accordingly (collaborative filtering method). It is also possible to combine both engines (3rd way: Hybrid Model of both Collaborative filtering and content-based approach) in order to build a more prosperous recommendation engine.

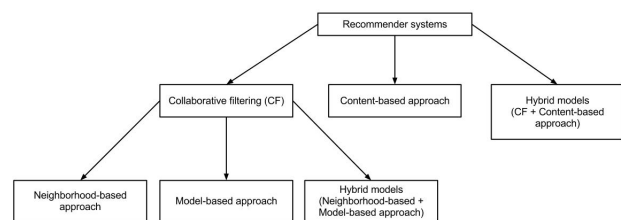


Fig. 1: Image separation

1. Content-based approach

Content-based filtering methods are based on a description of the item and a profile of the users preference. In a 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 (or is examining in the present). In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. In, short when I am logging on amazon,flipkart or any other e-commerce sites they recommend me the items which I purchased. So,their they are applying this kind of approach. Bu, here to show user some skills we have to check other users skills and on the basis of that we will suggest new user to gain this much of skills. So, thats why we cannot use this kind of apporch in our algorithm.

2. collaborative filtering approach

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). The underlying assumption of the collaborative filtering approach is that if a person A has the same opinion as a person B on an issue, A is more likely to have B's opinion on a different issue than that of a randomly chosen person. It means that if person A and B both are skilled in java and if person A gains skill in Matlab then our algorithm will recommend user B to gain skills in Matlab.

III. DATASET:

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: Candidate's Id, skills and Year

IV. MODULE 1:

A module that reads users profile and suggest a career path in terms of skillset – to be acquired.

Approach:

So, in this module our problem is to assign skills to particular person on the bases of all other persons's skills. For that firstly we have to gather the information about every person's skills. Then we have used Collaborative Filtering approach to assign skills to a person. Collaborative Filtering is a technique which tries to predict set of values from the prior experience or choice from many users and collaborating. Here, our data contains user's different skills and their experience in particular domain in terms of years. Our notations are like this:

$R(i, j)$: User j has gained i th skill.

$Y(i, j)$: experience of 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.

So initially 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. Our feature vector contains experience of particular person. Then we made a matrix X which is the average experience required for that particular job, where θ vector is a experience of each user. Now this becomes a classification problem and to solve that problem we have used linear regression method. Linear regression method's main approach is to minimize the cost function and minimize the mean square error. After running this algorithm for some iterations we will have the values of user matrix and experience matrix which will minimize

the cost function. Here, machine learning algorithm will be applied to update the weights in cost function. After all this we will have skills to be suggested by our algorithm to user. And we will use this approach to know which skills are required to us.

V. MODULE 2:

A module in which user enters a career goal and based on this career goal and other related information the platform suggest a career path.

Approach-1:

So, in this second module the user will write his/her desired career goal and on the basis of career goal our algorithm will suggest skills that person should achieve. It's an online version of first module. The steps are same in the module-2 as suggested in module-1. We have a 10×10 sparse matrix, its column contains numbers of users and rows contains numbers of skills. When new user comes we will assign required parameter X , θ and Y . After that we will calculate linear regression followed by gradient descent which will minimize cost function.

In our approach we just have to write new users id and its desired career goal and after the process we will give the list of skills to achieve.

Approach-2:

We have tried to implement second approach for the same module-2 problem where user has to enter his/her career goal and our algorithm will suggest skills to achieve. Here, the technology used to build our system is string matching, mean and fuzzy logic. The system will directly suggest user those skills which are achieved by most of the user. Even I have tried to use autocomplete recommendation where if a person writes System, he will be suggested career path having System as a key word.

Here, career goal is a javascript object. Every JS object contains job position and required skills. Job Position is taken as a Career Goal of that person which he/she wants to achieve. Then, the career goal is pushed into array for searching purpose and checking whether the same career goal is already present in the map or not. If career goal is already present then we will merge the skill set with given skills and will concat in previous skills of the same role. But if this is happening for the first time then add the skill corresponding to the new career goal. Then we remove all the duplicate values. So, ultimately we will have skills for our unique career goal. For doing auto-completion of career goal I am doing search route. Required skills for career goal will be returned by fetching career goal. Reduce function is going to map the skills set according to the number of times the given skill occurs. For example if for a particular career path, MATLAB comes for 4 times and R comes for 2 times then MATLAB will be the first suggested skill.

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 (Approach-1)

- 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. MAIN RESULT

A. Module-1

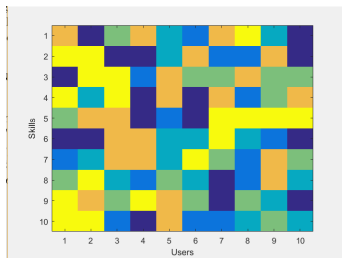


Fig. 2: User's and his skills

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

Fig. 3: Suggested Skills

B. Module-2 (Approach-1)

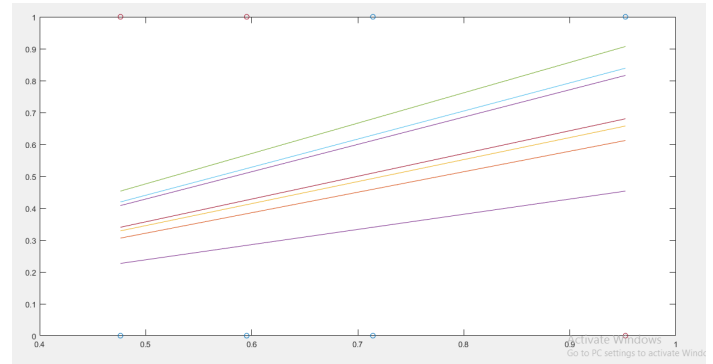


Fig. 4: Output of approach-1

C. Module-2 (Approach-2)

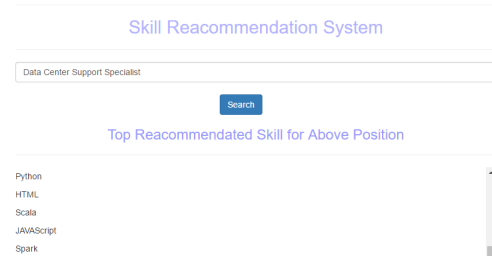


Fig. 5: Output of approach-2

VIII. CHALLENGES

First challenge we faced was that we were given data which contained so many parameters. So, we have to clean up the data and have data which only contains candidate's id, skills and experience. In practice, recommendation systems are based on large datasets. Here, the user-item matrix used for collaborative filtering is very large and sparse. So, another challenge we faced was that our matrix was very large. So, we have to first reduce the dimensions of matrix. And here for collaborative filtering the data should be very large. Because of that we faced some performance issues.

REFERENCES

- [1] https://en.wikipedia.org/wiki/Recommender_system
- [2] https://en.wikipedia.org/wiki/Collaborative_filtering
- [3] <https://www.coursera.org/learn/machine-learning/lecture/2WoBV/collaborative-filtering>