Algorithms and Optimization of Big Data : Final Exam Submission

Deep C. Patel (1401010),
School of Engineering and Applied Science, Ahmedabad University

***Abstract*—This paper aims to solve the problem of career-path recommendation system that was given to us as our final exam question. The problem is a recommendation system problem at its heart. We tackled this problem using dictionaries/hash tables and the matrices. First of all the module to clean the given data and extract some meaningful data was written. Then we scan other users' skills and based on that we suggest to the current user, the skills he must learn to pursue some career. There are two parts of this system, one which scans current user's skills and based on that suggests the possible career and additional skill-set to be acquired to pursue that career. In the second part the user enters the career he/she wants to pursue and we suggest the career path to him/her based on what skills he/she have currently and which skills are required. In first part we suggest the top closest careers in terms of minimum skills required to be gained to achieve them. Have successfully implemented the code in Python language on the given dataset.**

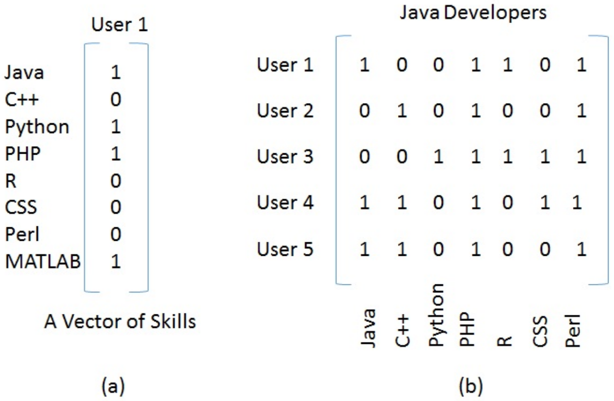
***Keywords*—Recommendation System, Dictionary, Hash Tables, Principal Components**

I. Introduction

The problem which was given to us as a part of the final exam was a career recommendation system and it has tremendous scope in the IT industries. All the companies want to retain the customers and they do so by giving them recommendation of the contents based on their taste to keep them engaged into their platform. One such problem is the career suggestion problem, though it is not intended for engaging users, it is a class of recommendation problem. Here, based on the user's current profile we have to suggest the relevant career to him/her and skill-set needed to be acquired in order to pursue that career. Most commonly the recommendation problems could be solved by content bases recommendation and collaborative filtering. Here we are going to apply content based filtering to solve this problem.

II. Framing the Problem Mathematically

During data extraction the "skill" attribute of each individual of some profession was extracted and represented as the vector shown in figure 1(a), where the absent skills are marked 0 and present skills are marked ones. Then for all such users these vectors were made and stacked into the matrix, where each matrix represented each profession as shown in figure 1(b). Now after this we will obtain PCAs of this matrix using SVD and reconstruct the data matrix with dominant PCAs only. Thus the dimensions of the data is reduced, but at the same time we get the essence of the whole data into reduced dimension form. Now we take the most dominant skills and write it to the file. We denote this process as data cleaning and extraction and it is illustrated in figure 2.



User 1

$$\begin{array}{l|c|} & \\ \text{Java} & 1 \\ \text{C++} & 0 \\ \text{Python} & 1 \\ \text{PHP} & 1 \\ \text{R} & 0 \\ \text{CSS} & 0 \\ \text{Perl} & 0 \\ \text{MATLAB} & 1 \end{array}$$

A Vector of Skills

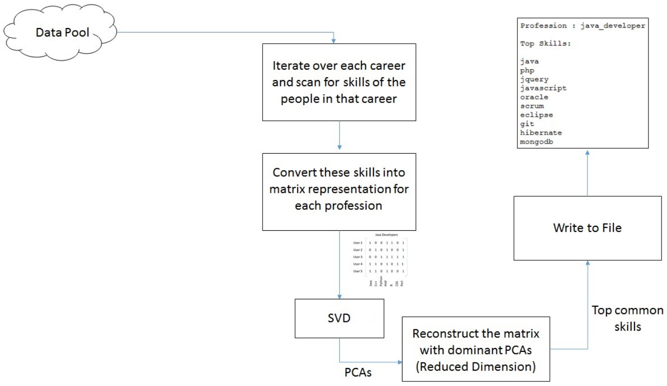
(a)

Java Developers

$$\begin{array}{l|ccccccc|} \text{User 1} & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ \text{User 2} & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ \text{User 3} & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ \text{User 4} & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ \text{User 5} & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ & \text{Java} & \text{C++} & \text{Python} & \text{PHP} & \text{R} & \text{CSS} & \text{Perl} \end{array}$$

(b)

Fig. 1: Vector and Matrix Representation of the Skills



Data Pool
Iterate over each career and scan for skills of the people in that career
Convert these skills into matrix representation for each profession
SVD
PCAs
Reconstruct the matrix with dominant PCAs (Reduced Dimension)
Top common skills
Write to File

```
Profession : java_developer

Top Skills:

java
php
jquery
javascript
oracle
scrum
eclipse
git
hibernate
mongodb
```

Fig. 2: Data Cleaning and Extraction

III. Recommending the Skills to the User

The task given to us was divided into two parts; 1). Provide the career path to the user in terms of skill-set to be acquired and 2). Give the career path to the user based on the career goal he enters. Thus we have made two modules each doing as stated respectively. After the preprocessing of the data we get the most important skills for each career, we use this information in making this modules. The generalized model for these modules is shown in figure 3.

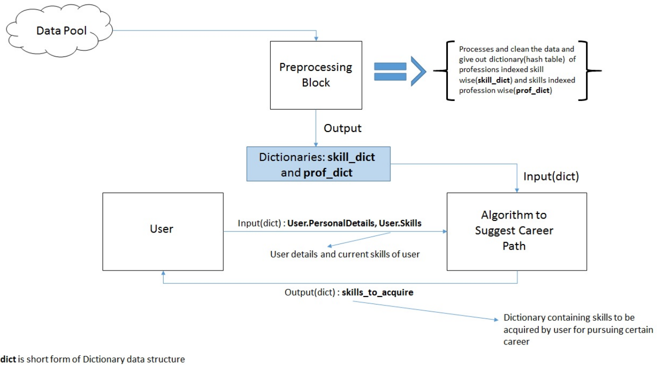
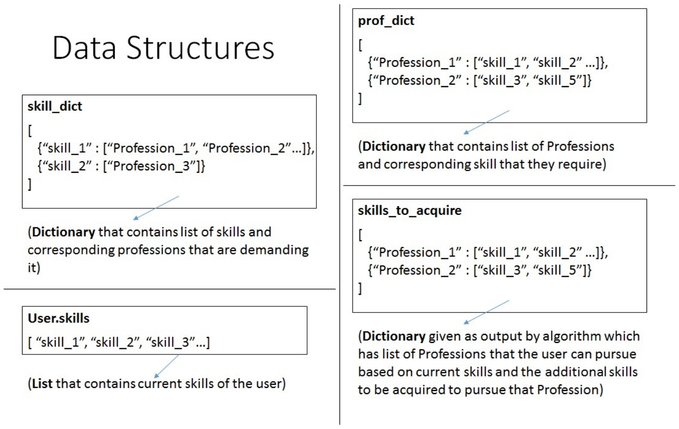

Data Pool
Preprocessing Block
Processes and clean the data and give out dictionary(hash table) of professions indexed skill wise(**skill_dict**) and skills indexed profession wise(**prof_dict**)
Output
Dictionaries: **skill_dict** and **prof_dict**
Input(dict)
User
Input(dict) : **User.PersonalDetails, User.Skills**
User details and current skills of user
Algorithm to Suggest Career Path
Output(dict) : **skills_to_acquire**
Dictionary containing skills to be acquired by user for pursuing certain career
*dict is short form of Dictionary data structure

Fig. 3: Generalized model for the modules

The description of the data structures used in the above generalized model is shown below:-



Data Structures

skill_dict
[
{"skill_1" : ["Profession_1", "Profession_2"...]},
{"skill_2" : ["Profession_3"]}
]

(**Dictionary** that contains list of skills and corresponding professions that are demanding it)

User.skills
["skill_1", "skill_2", "skill_3"...]

(**List** that contains current skills of the user)

prof_dict
[
{"Profession_1" : ["skill_1", "skill_2" ...]},
{"Profession_2" : ["skill_3", "skill_5"]}
]

(**Dictionary** that contains list of Professions and corresponding skill that they require)

skills_to_acquire
[
{"Profession_1" : ["skill_1", "skill_2" ...]},
{"Profession_2" : ["skill_3", "skill_5"]}
]

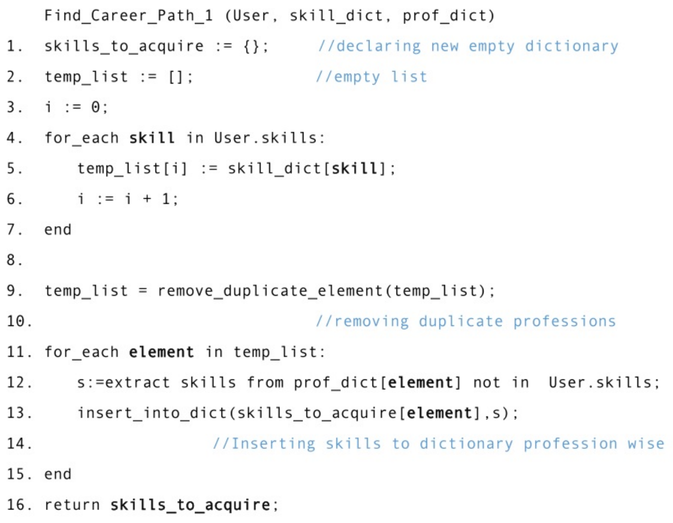
(**Dictionary** given as output by algorithm which has list of Professions that the user can pursue based on current skills and the additional skills to be acquired to pursue that Profession)

Fig. 4: Common data structures used in the model

Now the specialized algorithms for each module are discussed below, these could be fitted in the "algorithm" block of the generalized model in figure 3.

1) *Suggesting a career path in terms of skill-set to be acquired*

According to what we understand, the module is designed in such a way that it scans the current skills of the user and based on that decides which career the user should pursue so that he/she has to acquire minimum skill-set and can pursue it faster. After the career is decided the user is suggested the skills which he/she lacks, but need them to pursue that particular career. The dictionary or hash tables are extensively used in this algorithm to reduce the complexity of it. The pseudocode of the algorithm is shown in figure 5.

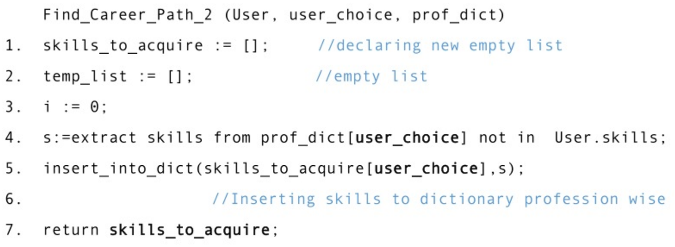


```
Find_Career_Path_1 (User, skill_dict, prof_dict)
skills_to_acquire := {};      //declaring new empty dictionary
temp_list := [];              //empty list
i := 0;
for_each skill in User.skills:
    temp_list[i] := skill_dict[skill];
    i := i + 1;
end

temp_list = remove_duplicate_element(temp_list);
                              //removing duplicate professions
for_each element in temp_list:
    s:=extract skills from prof_dict[element] not in  User.skills;
    insert_into_dict(skills_to_acquire[element],s);
                    //Inserting skills to dictionary profession wise
end
return skills_to_acquire;
```

Fig. 5: Pseudocode of the algorithm for module 1

2) *Suggesting a career path on basis of the career goal entered by the user*

Here the user enters the carrier goal and based on that we find out all the skills required to pursue that career using dictionary. Now, we will extract the skills for that career which the user don't have and will show to him/her. Additionally we will also show the user the duration in years for which he/she has to gain the experience. The duration we will show is the average experience people gain before joining that profession. We will also show the company which the user can join in his/her region after pursuing that career. We will also show the additional skills the persons working in the profession has, so that it makes task of finding the job easy for the user. The pseudocode for this algorithm is shown in figure 6.



```
Find_Career_Path_2 (User, user_choice, prof_dict)
skills_to_acquire := [];      //declaring new empty list
temp_list := [];              //empty list
i := 0;
s:=extract skills from prof_dict[user_choice] not in  User.skills;
insert_into_dict(skills_to_acquire[user_choice],s);
                    //Inserting skills to dictionary profession wise
return skills_to_acquire;
```

Fig. 6: Pseudocode of the algorithm for module 2

IV. Complexity Analysis of the Algorithms

1). *Preprocessing Algorithm*

–SVD :$O(n^3)$
–Scanning Files : O(No. of users)
Total : $O(n^3)$[here n is No. of skills

2). *Algorithm for Module 1*

–Let k = no. of skills with user
–Acquiring Careers in Loop 1 : $O(k)$
–Dictionary Query or Hashing : $O(1)$
–Duplicate Removal : O(k*no. of careers to be chosen)

–Skill to be acquired by user :

O((k - no. of skill for each career)*no. of careers chosen)

Total : O(k*no. of careers to be chosen)

3). Algorithm for Module 2

–Let k = no. of skills with user
–Skill to be acquired by user :

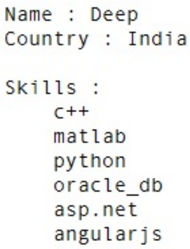
O((k - no. of skill for each career))

Total : O((k - no. of skill for each career))

V. Results

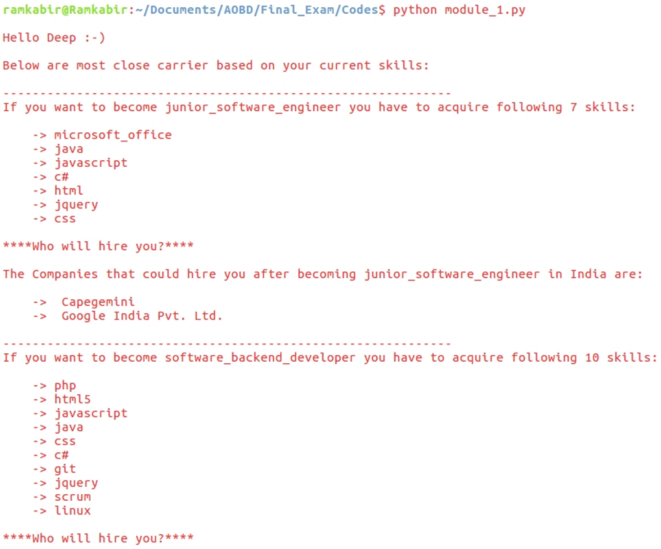
3) Module 1

Input :



```
Name : Deep
Country : India

Skills :
    c++
    matlab
    python
    oracle_db
    asp.net
    angularjs
```

Output :

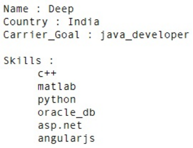


```
ramkabir@Ramkabir:~/Documents/AOBD/Final_Exam/Codes$ python module_1.py

Hello Deep :-)

Below are most close carrier based on your current skills:

------------------------------------------------------------
If you want to become junior_software_engineer you have to acquire following 7 skills:

    -> microsoft_office
    -> java
    -> javascript
    -> c#
    -> html
    -> jquery
    -> css

****Who will hire you?****

The Companies that could hire you after becoming junior_software_engineer in India are:

    ->  Capegemini
    ->  Google India Pvt. Ltd.

------------------------------------------------------------
If you want to become software_backend_developer you have to acquire following 10 skills:

    -> php
    -> html5
    -> javascript
    -> java
    -> css
    -> c#
    -> git
    -> jquery
    -> scrum
    -> linux

****Who will hire you?****
```

The input is the user data and skills that the user has and output is the career path in terms of the new skill-set to be acquired to pursue certain career and the companies that can hire the user if the user pursue that carrier.

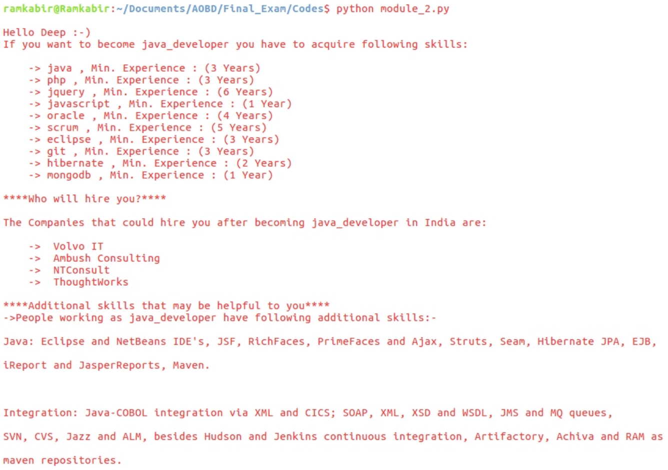
4) Module 2

Input :



```
Name : Deep
Country : India
Carrier_Goal : java_developer

Skills :
    c++
    matlab
    python
    oracle_db
    asp.net
    angularjs
```

Output :



```
ramkabir@Ramkabir:~/Documents/AOBD/Final_Exam/Codes$ python module_2.py

Hello Deep :-)
If you want to become java_developer you have to acquire following skills:

    -> java , Min. Experience : (3 Years)
    -> php , Min. Experience : (3 Years)
    -> jquery , Min. Experience : (6 Years)
    -> javascript , Min. Experience : (1 Year)
    -> oracle , Min. Experience : (4 Years)
    -> scrum , Min. Experience : (5 Years)
    -> eclipse , Min. Experience : (3 Years)
    -> git , Min. Experience : (3 Years)
    -> hibernate , Min. Experience : (2 Years)
    -> mongodb , Min. Experience : (1 Year)

****Who will hire you?****

The Companies that could hire you after becoming java_developer in India are:

    ->  Volvo IT
    ->  Ambush Consulting
    ->  NTConsult
    ->  ThoughtWorks

****Additional skills that may be helpful to you****
->People working as java_developer have following additional skills:-

Java: Eclipse and NetBeans IDE's, JSF, RichFaces, PrimeFaces and Ajax, Struts, Seam, Hibernate JPA, EJB,

iReport and JasperReports, Maven.


Integration: Java-COBOL integration via XML and CICS; SOAP, XML, XSD and WSDL, JMS and MQ queues,

SVN, CVS, Jazz and ALM, besides Hudson and Jenkins continuous integration, Artifactory, Achiva and RAM as

maven repositories.
```

The input is the user data and skills and the career goal that the user has and output is the career path in terms of the new skill-set to be acquired to pursue that career and the companies that can hire the user if the user pursue that carrier also we have mentioned that for how much time the user needs to get experience on certain skills before moving to industry to pursue career. The additional skills are the skills that are with the people who are currently working in the profession that the user is targeting to, this would help user to get overall picture of the field that he is planning to go.

VI. Github Code Link

Link :
https://github.com/deepcpatel/AOBD17_1401010/tree/master/Final_Exam_Submission

References

[1] http://infolab.stanford.edu/ ullman/mmds/ch9.pdf