# Algorithms and Optimization for Big Data-Final Exam Report

Akhil Vavadia(1401095)
School of Engineering And Applied Science,Ahmedabad University

***Abstract*—Now a days deciding career and skills required for the target career is very confusing as there are number of career and plenty of various skill for different careers. Job recommendation systems have been proposed in order to automate and simplify this task, also increasing its effectiveness. However, current approaches rely on scarce manually collected data that often do not completely reveal people skills. Our work aims to find out relationships between jobs and people skills making use of data user's profile(mainly skills).**

**Keywords:career/job position/profession , cosine similarity, Latent Semantic Analysis,SVD.**

## I. Introduction

It is always very difficult to decide career or job position or profession based on skills that we already have and its difficult task to gain knowledge about which skills are required or most important in particular profession.There are various models available which recommends jobs based on user attributes but there is very less work in this area of mapping between various professions and various skills which are importance in that profession.This analysis helps student and job seekers to decide their career progression path. Here in this paper we suggest two modules one for recommending jobs based on skills and other is recommending skills based on target career.

## II. Problem Statement

Here in this paper the problem we are trying solve is that,A company like linkedin wants to build a module that suggest its users the career progression path.When a user logs onto the platform, the platform reads user's profile and based on various parameters of this profile comes up with relevant suggestions on how the user should consider next set of skills to be acquired. Here we are focusing mainly on designing two modules given below.
*Module-1:*Reads user's profile and suggest a career path (in terms of skillset)to be acquired.
*Module-2:*User enters a career goal and based on this career goal and other related information the platform suggest a career path.

## III. Our Approach

From database we extracted mainly three data of job position(or profession),skills and user profile(consist of user current job position and skills).To represent data mathematically we represent it in vector forms, vector $U = \{u1, u2, ...\}$ of user profiles, vector $S = \{s1, s2, ...\}$ represents all distinct skills and vector $P = \{p1, p2, ...\}$ represents all distinct job positions.

From these vectors we make two matrices of weight matrix $w$ and and user skill matrix $U_s$.
$w$ matrix has all distinct skills as row and all distinct job position(professions) as its columns.so $w$ has dimensions $|S|$x$|P|$.

$$W_{i,j} = |u \epsilon U : s_i \epsilon S(u) \wedge p_j \epsilon P(u)|$$

$W_{i,j}$ is the number of user profile having both $s_i$ among skills and $p_j$ as position.So each element $c_{i,j}$ represent weight-importance of that $s_i$ skill in that $p_j$ job position.Here each column vector represents weighted skills , according to those possessed by persons employed in that position.

Matrix $U_s$ has all distinct skills as row and all user as its columns so it has dimension $|S|$x$|U|$.It is binary matrix with 1's at skills that user have in his profile from all skills and 0's at skill which are not in user profile.

### A. Module-1

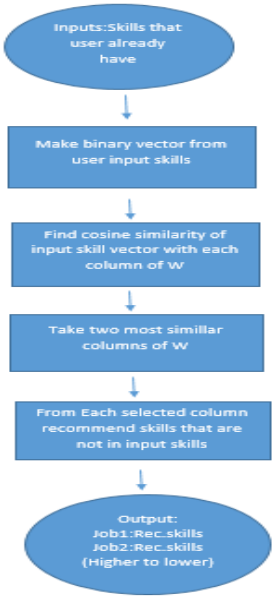


Fig. 1: Module-2

Cosine Similarity:The cosine similarity between two vectors is a measure that calculates the cosine of the angle between them. This metric is a measurement of orientation and not

magnitude, it can be seen as a comparison between vectors on a normalized space because we're not taking into the consideration only the magnitude of each vector, but the angle between the vectors.Two similar vectors might have higher euclidean distance but when we consider both magnitude and direction we get actual similarity between vectors. .cosine similarity equation is,

$$cos(\theta) = \frac{x \cdot y}{||x|| \cdot ||y||}$$

Algorithm:
step-1:Take skills that user already has as input and make binary vector($U_{s-input}$) which has dimension $|S|$x1 , placing 1's at skills that user already has and 0's everywhere else.
step-2:Calculate cosine similarity of $U_{s-input}$ with each column vector of $W$,which gives us how particular job position has similar skills as user's.
step-3:Recommend k most similar job position to user.
step-4:Now for each recommended job position($Rec_1, Rec_2, ..., Rec_k$),which is a vector of weighted skills(extracted from columns of $W$)place 1 in place of weights and subtract it from $U_{s-input}$ which gives us vector with all required skills for that job position that user does not have.
step-5:Recommend skills to be acquired for recommended job position in Higher to lower preference order.

### B. Module-2

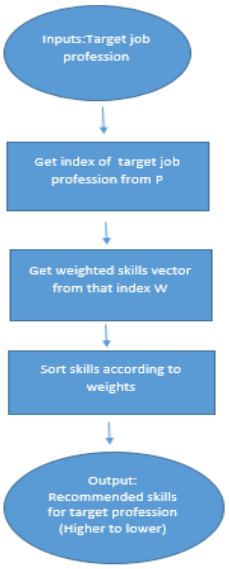


Fig. 2: Module-2

Algorithm:
Step-1:User give his target career goal(profession) as input.
Step-2:Find index of target profession from positions vector $P$ using string matching.
Step-3:From obtained index extract column vectors at that index position in $W$ matrix.So we obtained weighted skill vector of target profession.
Step-4:Sort skills in descending order of weights.
Step-5:Recommend skills from higher weight to lower weight for targeted profession.

## IV. Implementation and Results

### A. Data Composition

From given database we have extracted in three data files which are,

- skills.xlsx-contains all distinct skills
- positions.xlsx-contains all distinct positions
- userprofile.xlsx-contains user's current position and skills he currently has.

### B. Implementation On Smaller Dataset

We implemented both algorithms on smaller sample of data to understand both $W$ and $U_s$ matrices.

| skills | Software Engineer | Software Developer | Design Engineer | Automation Engineer |
|---|---|---|---|---|
| java | 2 | 1 | 0 | 0 |
| python | 1 | 1 | 0 | 0 |
| c | 1 | 1 | 0 | 0 |
| matlab | 0 | 0 | 0 | 0 |
| testing | 0 | 0 | 0 | 0 |
| sql | 1 | 0 | 0 | 0 |
| automation | 0 | 0 | 0 | 1 |
| animation | 0 | 0 | 1 | 0 |
| graphic desig | 0 | 0 | 1 | 0 |
| autocad | 0 | 0 | 1 | 0 |

Fig. 3: $W$ matrix

| skills | Candidate-1 | Candidate-2 | Candidate-3 | Candidate-4 | Candidate-5 |
|---|---|---|---|---|---|
| java | 1 | 1 | 0 | 0 | 1 |
| python | 1 | 0 | 0 | 0 | 1 |
| c | 0 | 1 | 0 | 0 | 1 |
| matlab | 0 | 0 | 0 | 0 | 0 |
| testing | 0 | 0 | 0 | 0 | 0 |
| sql | 0 | 0 | 0 | 0 | 0 |
| automation | 0 | 0 | 0 | 1 | 0 |
| animation | 0 | 0 | 1 | 0 | 0 |
| graphic desig | 0 | 0 | 1 | 0 | 0 |
| autocad | 0 | 0 | 1 | 0 | 0 |

Fig. 4: $U_s$ matrix

### C. Results

```
Enter your skills:java,c
Most recommended job position for user is software developer:
for this job position skills to be acquired are(High to low):
    'python'

Second most recommended job position for user is software engineer
for this job position skills to be acquired are (High to low):
    'python'

    'sql'
```

Fig. 5: Module-1 output

### D. Proof of Correctness

We can check our module-1 algorithm if we give complete input skills of user which we already have in database then our algorithm should return user's current job position as first recommendation and any other profession with all the skills required as second recommendation.

```
>> module2
Enter the profession:software engineer
Recommended skills(High Preference-->Low Preferemce):
    'java'    'python'    'c'    'sql'

>> module2
Enter the profession:design engineer
Recommended skills(High Preference-->Low Preferemce):
    'graphic design'    'animation'    'autocad'
```

Fig. 6: Module-2 output

```
>> module1
Enter your skills:graphic design,animation,autocad
Most recommended job position for user is design engineer:
for this job position skills to be acquired are(High to low):
Second most recommended job position for user is software engineer
for this job position skills to be acquired are (High to low):
    'java'

    'python'

    'c'

    'sql'
```

Fig. 7: Module-1 POC

We have one entry in our database as user's current positions is design engineer and his skills are graphic design,animation,autocad.Now as in above fig.7 when we give these three skills as input it gives most recommended profession as design engineer with no more skill requirements and as second recommendation it gives software engineer with all its skills.

For module-1 the recommended skills for targeted profession should be sorted list of skills from column of that profession $W$ matrix.That can be checked by $W$ matrix in Fig.3 and output of module-1 which is in Fig.5.

## V. Future work

In this paper we have assumed that there no spelling error in data and input but in future we can achieve this using levenshtein distance algothms for aproximate string matching.

Also there are various nearly similar professions for example software engineer,software developer,java developer,etc. so we can make cluster of this similar prfoession so we can accurately recommend skills for that professions.This can be achieved by Latent semantic analysis which we can implement by performing SVD on our W matrix and take first k eigen valuse only.

## References


[1] http://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/

[2] Job Recommendation From Semantic Similarity of LinkedIn Users' Skills, Conference Paper.January 2016, DOI: 10.5220/0005702302700277