Subject Recommendation System

CMPE-256 Individual Project

Maulin Bodiwala (SID: - 013733590)

1. Abstract

Choosing a right Subject in formative years is very important decision as student's future depends on this one decision. Student by himself is not experienced enough to take right decision in his earlier years of college and end up getting bad scores. Selecting wrong Subjects means mismatch between student's aptitude, capability and personal interest. Since there is no other reliable source generally available that can guide the student towards the most suitable direction, recommendation system has been evolved to provide him/her guidance in selecting a right Subject. Thus, the project idea is to develop a system for helping the students to choose an elective Subject which would be best suited for him/her based on features like previous performance, skill set, interest, ability to learn, etc. To execute this, I planned to take few inputs from the user and with the help of an algorithm which would combine all the features and in result output a ranked list of Subjects with the expected grades in each subject which he/she should proceed with. I have used Collaborative based filtering, content based filtering, and singular value thresholding algorithm for kernel matrix completion for predicting grades of elective Subjects.

2. Motivation

There have been a number of researches conducted in the field of educational data mining to predict the performance of student. In this sequence, our project discusses the problem of elective subject selection and proposes a solution through a recommendation system which may help student to make the right choice which best suites his/her. I am trying to provide a solution to this problem using the latest technologies and researches on recommendation system. Recommendation system enhance the teaching and learning, recommend good solution, analyze data and offer data to modify activity plan. In the last few years recommendation system have provided valuable solution in opting one choice out of available many choices by focusing on logical relationship. So that people behave intelligently and make the right choice. Student assessments are traditional method to predict student performance such as failing or passing or forecasting successful completion of the Subject, in this continuation predicting the classification of degree or achievement.

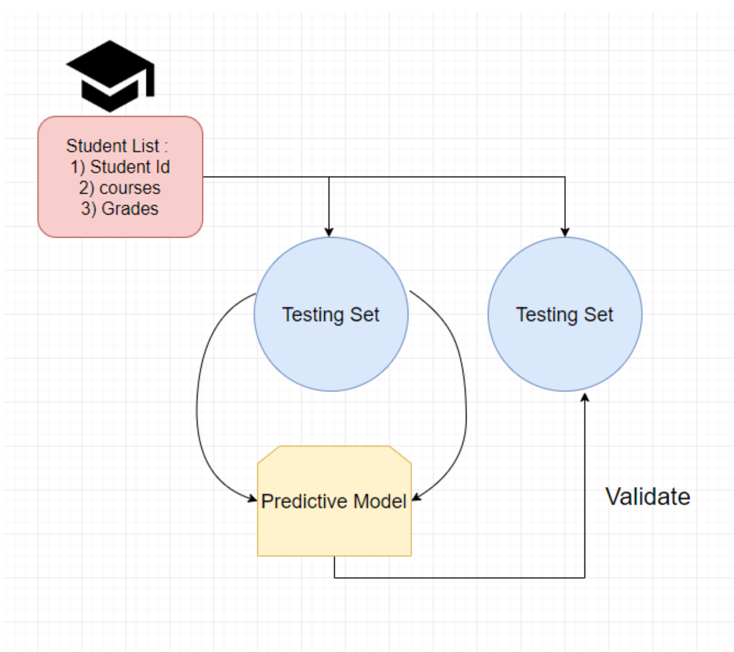
3 Literature Survey

Through the various research papers first the concept of recommendation system and its applications was understood. I got to know the various kinds of recommendation systems content-based, collaborative and hybrid and all these in the case of Book recommendation gave same results. Then I moved on to Subject recommendation and learnt about various techniques like ACO (Ant Colony Optimization), MLbased approaches like regression, SVM, clustering,etc. The best motivation was provided by the paper which provided us with 4 algorithms to predict the Subject a student should opt for. After experiment and result analysis, it is found that among 4 association rule algorithms 'Apriori Approach' performs the best with better accuracy in Subject recommender system.

Firstly, after reading various research papers on recommendation system, I understood the idea, types, and working of it. In those papers various methods were discussed and experiments were conducted with real datasets to assess the overall performance of the proposed approach. They have used collaborative recommendation system that employed association rules algorithm to recommend university elective Subjects to a target student based on what other similar students have taken. Also, they have calculated the faculty expertise because faculty also plays a great role as they teach those subjects.

4. System Design

The Recommendation system I have proposed takes input from the students regarding their interests and future goals. The system also takes the information about the past grades, prerequisites and evaluation criteria of Subjects from the 3 data set provided to the system. The system tries to predict the grades of students in the elective Subjects offered, using various techniques like user-based collaborative filtering, item based collaborative filtering, kernel-based threshold mechanism and content-based approach. Then it compares the results and uses the method giving least error. The output of the system is a ranked list of Subjects according to the input given by the student along with the predicted grades by the above-mentioned method.



Data Set and Preprocessing: -

```
[4]: dataValues = np.load("../data_raw_05_15.npy" , encoding = "latin1")
     print(dataValues)
```

```
                    0                                1    2    3      4     5
0   TH301                      Đặc tả hình thức  NaN    1      V    0.0
1   TH900                    Luận văn tốt nghiệp  NaN   TN    9.9    9.9
2   TH160                       Thực tập thực tế  NaN   T1     10   10.0
3   TH604                          An toàn mạng  NaN    T   10.0   10.0
4   TH407           Mã hóa thông tin & ứng dụng  NaN    1   10.0   10.0
5   TH304                             Nhận dạng  NaN    1    8.0    8.0
6   TH401           Xây dựng PM hướng đối tượng  NaN    1   10.0   10.0
7   TH138                      Xử lý tín hiệu số  NaN    1    6.0    6.0
8   AN115                             Anh văn 5  NaN   25   8.30    8.5
9   TH608                Nhập môn mã hóa & mật mã  NaN   T1   10.0   10.0
10  TH405  Phân tích, thiết kế hướng đối tượng  NaN   T1    9.0    9.0
11  TH117                Quản lý đồ án phần mềm  NaN   T1   10.0   10.0
12  TR040                  Tư tưởng Hồ Chí Minh  NaN    2    8.0    8.0
13  TH303                              Xử lý ảnh  NaN   T1    9.5    9.5
14  AN114                             Anh văn 4  NaN    2   9.10    8.5
15  TH609            Automat & ngôn ngữ hình thức  NaN   TN   6.50    6.5
16  TH602       Các công nghệ lập trình hiện đại  NaN   TN  10.00   10.0
17  TH110                     Công nghệ phần mềm  NaN   TN  10.00   10.0
18  TH109                          Đồ họa máy tính  NaN   TN     10   10.0
```

After Preprocessing:

Pre-processing

```
In [10]: # This block inits data of these 3 dicts
course_namedict = {} # Convert from course id to course name
course_ids = {} # convert from course id to index in range [0, n_items)
user_ids = {} # convert from student id to index in range [0, n_users)

uid = 0
for student in dataValues:
    studentMssv = student[0]
    student_table_score = student[1]
    for scoreidx in range(0, len(student_table_score)):
        course_id = student_table_score[0][scoreidx]
        if course_id in course_convert_id:
            course_id = course_convert_id[course_id]
        course_namedict[course_id] = student_table_score[1][scoreidx]
    if np.sum(student_table_score[5]) != 0:
        user_ids[studentMssv] = uid
        uid += 1

cid = 0
for course_id in course_namedict.keys():
    course_ids[course_id] = cid
    cid += 1

print(user_ids, course_ids)
```

```
{'0512001': 0, '0512002': 1, '0512003': 2, '0512004': 3, '0512005': 4, '0512006': 5, '0512007': 6, '0512008': 7, '0512009':
8, '0512010': 9, '0512011': 10, '0512012': 11, '0512013': 12, '0512014': 13, '0512015': 14, '0512016': 15, '0512017': 16, '05
12018': 17, '0512019': 18, '0512020': 19, '0512021': 20, '0512022': 21, '0512023': 22, '0512024': 23, '0512025': 24, '051202
6': 25, '0512027': 26, '0512029': 27, '0512030': 28, '0512031': 29, '0512032': 30, '0512033': 31, '0512034': 32, '0512035': 3
3, '0512036': 34, '0512037': 35, '0512038': 36, '0512039': 37, '0512040': 38, '0512041': 39, '0512042': 40, '0512043': 41, '0
512044': 42, '0512045': 43, '0512046': 44, '0512047': 45, '0512048': 46, '0512050': 47, '0512051': 48, '0512052': 49, '051205
3': 50, '0512054': 51, '0512055': 52, '0512056': 53, '0512057': 54, '0512058': 55, '0512059': 56, '0512060': 57, '0512061': 5
8, '0512062': 59, '0512063': 60, '0512064': 61, '0512065': 62, '0512066': 63, '0512067': 64, '0512068': 65, '0512069': 66, '0
512070': 67, '0512071': 68, '0512072': 69, '0512073': 70, '0512074': 71, '0512075': 72, '0512076': 73, '0512077': 74, '051207
8': 75, '0512079': 76, '0512080': 77, '0512081': 78, '0512082': 79, '0512083': 80, '0512084': 81, '0512085': 82, '0512086': 8
3, '0512087': 84, '0512088': 85, '0512089': 86, '0512090': 87, '0512091': 88, '0512092': 89, '0512093': 90, '0512094': 91, '0
512095': 92, '0512096': 93, '0512097': 94, '0512098': 95, '0512099': 96, '0512100': 97, '0512101': 98, '0512102': 99, '051210
```

In [7]:

```python
nusers = len(user_ids)
nitems = len(course_ids)
print ("Number of users:", nusers)
print ("Number of items:", nitems)
```

```
Number of users: 5551
Number of items: 278
```

[11]:

```python
# This block creates a score matrix which is the input of CF models
matrix_data = np.zeros((nusers, nitems))
i=0
for user in dataValues:
    [mssv, scoretable] = user
    if mssv not in user_ids:
        continue
    for user_courseidx in range(0, len(scoretable)):
        course_id = scoretable[0][user_courseidx]
        course_id = course_convert_id[course_id] if course_id in course_convert_id else course_id
        user_coursescore = scoretable[5][user_courseidx]
        if user_coursescore != 0 and not math.isnan(user_coursescore):
            matrix_data[user_ids[mssv], course_ids[course_id]] = user_coursescore
            i+=1
print(i)
print (matrix_data)
```

```
201186
[[ 0.   9.9 10.  ...  0.   0.   0. ]
 [ 0.   0.   0.   ...  0.   0.   0. ]
 [ 0.   0.   0.   ...  0.   0.   0. ]
 ...
 [ 0.   0.   0.   ...  7.5  0.   0. ]
 [ 0.   0.   0.   ...  0.   0.   0. ]
 [ 0.   0.   0.   ...  4.   0.   0. ]]
```

5 Method

5.1 Singular Value Thresholding Algorithm

The first method is a simple rst-order and easy-to-implement algorithm that is extremely efficient at addressing problems in which the optimal solution has low rank. The algorithm is iterative and produces a sequence of matrices Xk , Yk and at each step, mainly performs a soft-thresholding operation on the singular values of the matrix Yk .

The singular value thresholding algorithm for approximately solving the nuclear norm minimization problem and by extension, problems of the form

$$\text{minimize } ||X||_*$$

$$\text{subject to} A(X) - b$$

where Ai s a linear operator acting on the space of n1 × n2 matrices and b ∈ Rm. This algorithm is a simple rst-order method, and is especially well suited for problems of very large sizes in which the solution has low rank.I sketch this algorithm in the special matrix completion setting and let P be the orthogonal projector onto the span of matrices vanishing outside of so that the (i, j) th component of P(X) is equal to Xij if (i, j) ∈ Ω and zero otherwise.

```
7]: prediction = compute_predict(matrix_data, user_similarity)
    print(prediction)

[[8.62702461 8.9364805  8.61799187 ... 8.45714286 8.45714286 8.45714286]
 [7.31794732 7.52053893 7.30328016 ... 7.25       7.25       7.25      ]
 [4.87531272 5.04905367 4.83513717 ... 4.75111752 4.76315789 4.76286936]
 ...
 [7.59106994 7.59142325 7.59122135 ... 7.32266848 7.59090909 7.58932142]
 [6.50027139 6.5004805  6.5005979  ... 6.25742135 6.5        6.49869358]
 [3.81267973 3.81278999 3.81259433 ... 3.52990346 3.8125     3.81068141]]
```

5.2 Content Based Algorithm

A content-based recommendation system recommends an item to a user based upon a description of the item and a profile of the user's interests. Content based recommendation systems may be used in a variety of domains ranging from recommending web pages, news articles, restaurants, television programs, and items for sale. Although the details of various systems differ, content based recommendation systems share in common a means for describing the items that may be recommended, a means for creating a profile of the user that describes the types of items the user likes, and means of comparing items to the user profile to determine what to recommend. The profile is often created and updated automatically in response to feedback on the desirability of items that have been presented to the user.

5.3 Grade Prediction method

The method predicts student grade in elective Subject from grades obtained in prerequisites subjects and average performance of student in all Subjects taken. 5 The data set consist of four things i.e. Student, Elective, Subject (which constitute the elective subject) and weight(assigned to subject for a elective). The obtained marks in prerequisite subject will be used for the purpose of finding the score of student in respective elective. Student elective score: To find the score of elective following formula will be used

$$ElectiveScore = \sum_{i=0}^{n} S_i * W$$

Where i indicate subject number Si indicate obtained score grade in the ith subject Wi indicate weight of ith subject in curriculum Using this formula score of each elective is calculated. After multiplying subject percentage score with respective weight value of each subject which constitute the elective curriculum, sum all of them. This data table is used to recommend the most suitable elective subject for the student.

5.4 Collaborative filtering:

5.4.1 User Based Approach

The steps followed in user-based CF to make a prediction for Student St are as follows:

Step 1: Similarity between the target student St and every other student is calculated.

Step 2: Based on their similarity value with student St , set of k students, most similar to target student St is then selected.

Step 3: Finally, prediction of grades for a student St in Subject It is generated by taking the weighted average of the grades scored by the k similar students to Subject it. For step 1 there are many algorithms but Pearson -r correlation coefficient performs the best. Using this I calculated sim(u,v) i.e. similar value of user u with v.

$$Sim_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r_u})^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r_v})^2}}$$

Here ru,i denotes the rating of user u for item i, and ru is the average rating given by user u calculated over all items rated by u. Similarly, rv denotes the rating of user v for item i, and rv is the average rating given by user v calculated over all items rated by v. Now for last step, grades of target student is calculated using an adjusted weightage sum formula, to take into account the fact that different students have different grades distributions.

$$P_{u,j} = \bar{r_u} + \frac{\sum_{v \in V} Sim_{u,v}(r_{v,i} - \bar{r_v})}{\sum_{v \in V} |Sim_{u,v}|}$$

In [22]:
```
mean_userscore = np.true_divide(train_matrix.sum(1),(train_matrix!=0).sum(1))
u, s, vt = svds(np.subtract(train_matrix, mean_userscore[:, np.newaxis], where=train_matrix!=0), k = 9)
s_matrix=np.diag(s)
train_prediction = np.dot(np.dot(u, s_matrix), vt) + mean_userscore[:, np.newaxis]
print ('Root Mean Squared Error: ' + str(rmse_compute(train_prediction, test_matrix)))
train_prediction_0 = np.dot(np.dot(u[0, :], s_matrix), vt)
print(train_prediction_0 )
```

```
Root Mean Squared Error: 1.6533213598214342
[ 5.79907262e-02  1.40629135e-01  3.44401752e-02  1.33471258e-03
  4.56213305e-02  9.89082238e-03  5.47278737e-02  1.24359633e-03
 -1.65550348e-01  1.14601632e-02  4.98164577e-03  1.75804255e-01
 -2.65017466e-02  6.76491616e-03 -1.42821635e-01  3.64906557e-03
  1.72707017e-01  2.13033449e-01 -9.05480924e-02  2.70726571e-02
  3.36872241e-02  1.03479501e-01 -1.04411314e-01 -6.63628600e-02
 -1.53312374e-01 -2.06370488e-01  9.59595596e-02 -2.75644104e-03
  5.54256562e-02 -9.23484941e-02 -2.94467699e-02 -1.70144931e-01
 -1.70221641e-02 -2.81984014e-01  8.51398774e-02 -4.43949669e-01
 -9.50909866e-03 -5.43411389e-03 -1.78806449e-01 -1.84702813e-01
 -5.17918136e-02 -2.17229323e-01  2.28462931e-02 -2.24064141e-01
 -1.12896395e-01 -7.51211692e-02 -7.60412526e-02 -1.26114871e-01
  3.69253713e-02 -7.87607359e-03 -4.67651470e-02  2.78876716e-02
 -3.62857769e-02 -1.40612800e-02 -5.38263896e-03 -8.66649680e-02
  2.66000971e-02 -1.55850633e-03  5.39976297e-02  1.28529658e-03
  1.14552379e-02  9.94307295e-02 -5.22569407e-02  3.98990674e-02
  1.16042330e-02 -2.43419655e-02  9.33831548e-02 -4.68722977e-03
  5.48984582e-04  1.87661536e-02  8.24504504e-03 -1.28216592e-03
  1.38134016e-01  3.75383791e-02  1.50713942e-02  1.40132972e-02
  1.31289927e-02  1.60719638e-01  6.56556619e-02  4.95131338e-03
```

5.4.2 Item Based

The main difference between item-based CF and user-based CF is that itembased CF generates predictions based on a model of item-item similarity rather than user-user similarity. In item-based collaborative filtering, first, similarities between the various Subjects are computed. Then from the set of Subjects previously taken (score) by the target user, k Subjects most similar to the target Subject are selected. For computing the prediction for the target Subject, weighted average is taken of the target student's scores on the k similar Subjects earlier selected. Let the set of students who have scores of both Subjects i and j be denoted by U, then similarity coefficient (Simi,j) between them is calculated as

$$Sim_{i,j} = \frac{\sum_{u\in U}(r_{u,i}-\bar{r_u})(r_{u,j}-\bar{r_u})}{\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r_u})^2}\sqrt{\sum_{u\in U}(r_{v,i}-\bar{r_v})^2}}$$

Here ru,i denote the scores of student u for Subject i, and ru is the average scores scored by user u calculated over all Subjects previously taken by u. Similarly, ru,j denotes the rating of student u for Subject j. To compute the predicted rating for a target item i for target user u, I use the following formula.

$$P_{u,j} = \frac{\sum_{j\in I} Sim_{i,j} * r_{u,j}}{\sum_{j\in I} |Sim_{u,v}|}$$

```
train_matrix, test_matrix = test_train_split(matrix_data)
```

```
train_prediction = compute_predict(train_matrix, user_similarity)
print(train_prediction)
```

```
[[8.57599474 8.84431495 8.58742924 ... 8.46153846 8.46153846 8.46153846]
 [7.29855105 7.46540612 7.28962635 ... 7.25       7.25       7.25      ]
 [4.70241206 4.85567986 4.68096929 ... 4.61662614 4.625      4.625     ]
 ...
 [7.59107943 7.5913028  7.59114641 ... 7.34635364 7.59090909 7.59090909]
 [6.50027772 6.50046956 6.50027772 ... 6.27766504 6.5        6.5       ]
 [3.81270478 3.81268877 3.81259924 ... 3.55397722 3.8125     3.8125    ]]
```

```
print ('Rooted mean squared error: ' + str(rmse_compute(train_prediction, test_matrix)))
```

```
Rooted mean squared error: 1.6721448563770709
```

6 Experiment

6.1. For Grade Prediction

The first experiment involves completion of the matrix which has entries of students grades in particular Subjects. The above methods have been applied on the matrix to find the most efficient method through the process of cross validation.

For grades prediction of a Subject having prerequisites, Subjects required prerequisites are identified for the semester. Since here prediction is based only on student's own performance, there is no need of hiding records of other student it will not affect the predicted result. I used the average performance of a student and performance in prerequisite Subjects values from dataset. Each prerequisite Subject is given a weight which range from 0 to 1 representing the importance of prerequisite Subject to take present Subject.

6.2. For Listing Subjects

For rank list of Subjects based on evaluation criteria and student's preference of Subject structure, content-based approach is used. The Subject score for each Subject is calculated, then the Subjects which are above the average score of all Subjects are recommended ranked by their scores. Profiles are created for both student and Subject representing student interest and Subject attribute. Here, also prediction is based only on student's own preferences and Subject credit structure, there is no need of hiding records of other student it will not affect the predicted rank list of Subjects.

7. Result

Singular Value Thresholding Algorithm - Since Data set for item is less i.e. 18 Subjects and 77 students so for kernel matrix completion between different Subjects comes to out in between. Using these Subject predictions of grades for student deviated maximum to positive deviation and negative deviation for few students, average deviation is. Hence, It shows that this technique gives decent result for the given data set. Item Based Algorithm - Since Data set for item is less Subjects and 77 students so for similarity comparison between different Subjects comes to out in between. Using, these Subject predictions of grades for student deviated maximum to positive deviation and -2.9 negative deviation for few students, average deviation is. Hence, It shows that Item Based Collaborative approach performing poor in given dataset, because of less no of k-neighbors (similar Subjects). User Based Algorithm - Since here I am calculating similarity between students which are large in numbers than Subjects, it is giving better accuracy in predicting Grades. Graph given below includes maximum positive deviation, maximum negative deviation and average deviation with different no of k-neighbors. It shows that when data or no of k-neighbors (similar users) is reduces deviation from actual value is increases. In Subjects which do not have prerequisite have maximum deviation of +3 to -3 of actual grade obtained, this is because grade predicted is average score of students till now which may vary much for student. For other Subjects 8 maximum deviation of -2 to +2 of actual grade obtained, and average deviation is -1 to +1.

8. Conclusion and Future Work

The above experiments show that User Based Collaborative performs better than Item Based Collaborative Filter and Singular Value Thresholding Algorithm for grade prediction for available amount of data. The efficiency of the system can be further improved by increasing the data set to get better results like for comparison I can include all student's record so far studying in the particular Institute. For Subject's ranked list, system can be developed by including different modules like based on placements and faculty etc. As these parameters also play an important role in the decision of the Subject selection.