Course Recommendation System

Report submitted in partial fulfillment of the requirement for the degree of

B.Tech

in

Computer Science & Engineering

ANd9GcRcmIsloy3g35B_JsRNO8FT1UVoHApPujW-ZfaOSnKNCyg35wmufV3iEg

Under the supervision

of

Ms. Charu Gupta

Assistant Professor

Department of Computer Science & Engineering

by

Achintya Sarkar 00720802715 , CSE-A

Himanshu Ratnani 03220802715 , CSE-A

Aayush Tiwari 00420802715 , CSE-A

Abhishek Jain 0062002715, CSE-A

Department of Computer Science & Engineering

Bhagwan Parshuram Institute of Technology

PSP-4, Sec-17, Rohini , Delhi-89

March 2019

1.Introduction

Nowadays, the amount of educational resources spread on Internet is huge and diverse. Massive Online Open Courses (MOOCs) such us Coursera, Udacity, EdX, to name a few, are gaining momentum[3] . It is possible to find courses from almost every knowledge domain[1]. This vast offer overwhelm any user willing to find courses according to his/her background.

In this project we propose a system for online courses recommendation, although MOOCs courses are primarily focused. To do so, we rely on course reviews, NLP [4] and we use a set of tools for using sentiments and classify courses based on them. Accordingly each course is assigned to a set of users who get a particular scores based on correlation [2]. Each document is a combination of topics and each topic is a probability distribution over words. Topic models are a type of graphical model based on Bayesian networks or SVM Classifiers [1]. The generative process described by a topic model does not make any assumptions about the order of words as they appear in documents. The only information relevant to the model is the number of times words are produce, this is known as the “bag-of- words” assumption [1].

1.1 Recommendation System

A recommender system or a recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. Recommender systems are a useful alternative to search algorithms since they help users discover items they might not have found otherwise. Recommender systems typically produce a list of recommendations in one of two ways – through collaborative filtering or through content-based filtering. Collaborative filtering approaches build a model from a user's past behaviour and as well as similar decisions made by other users. This model is then used to predict items or ratings for items that the user may have an interest in. Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties and these approaches are often combined by Hybrid Recommender Systems.

Mostly used in the digital domain, majority of today’s E-Commerce sites like eBay, Amazon, Alibaba etc make use of their proprietary recommendation algorithms in order to serve better the customers with the products they are bound to like.

1.2 *Types of Recommendation System*

1.2.1Collaborative Filtering

Collaborative filtering, also referred to as social filtering, filters information by using the recommendations of other people. 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.

1.2.2Content based filtering

These filtering methods are based on the description of an item and a profile of the user’s preferred choices. In a content-based recommendation system, keywords are used to describe the items; besides, a user profile is built to state the type of item this user likes. The algorithms try to recommend products which are similar to the ones that a user has liked in the past.

1.2.3 Hybrid recommendation system

Recent research shows that combining collaborative and content-based recommendation can be more effective. Hybrid approaches can be implemented by making content-based and collaborative-based predictions separately and then combining them. Further, by adding content-based capabilities to a collaborative-based approach and vice versa; or by unifying the approaches into one model.

2. Problem Statement

To get the data from various data sources such as Kaggle, UCL etc. To analyze the reviews using sentiment analysis which involves generating a classifier and testing it. To recommend online courses to users according to the courses suggested to the similar users before. It is quite a difficult task for a student to be able to take wise decisions about the courses he has to take up specially, if he is a new student and has few contacts to refer to. Even if he did have the contacts it would be subjective to a few opinions, which might not ensure the quality of the decision. Moreover this process could result in students choosing courses that might not be totally suited to their liking as it is hard for them to assess the course based on a few opinions. Taking a course without much value to the student would mean monetary wastage as well as wastage of time both of which are very crucial resources to a student.

2.1 Need for Project

The proposed project suggests an optimum course for each student separately. This gives each user the flexibility to try new course which they will be able to complete within stipulated time and gain knowledge without having to worry if the suggested course would neither be too easy nor too difficult for them. This system also saves the time of user which could have been lost due to sheer number of online courses available on the internet.

3. Design Phase

3.1 . Making a sentiment classifier for getting a uniform Score from new Reviews

Step 1 : Get the review DataSet containing the user reviews , the course name , a rating parameter .

Step 2 : To get a random set of course data , shuffle the dataset using *sklearn.utils.shuffle*

Step 3 : Preprocessing the dataSet .

a) Removing Non english reviews.

b) Reindexing the dataset after shuffling . Step 4 : Making the featureSet .

a) Select most common 3000 words as features .

b) Make a 2d matrix with above selected words as keys and True False as value for each review .

Step 5 : Train the NaiveBayesClassifier using the above created feature sets .

Step 6 : For any new review generate the featureSet and pass it in the classifier for predicting the score

END  
3.2. Making the Recommendation System

Step 1 : Get a User Item Matrix with Each user giving reviews about the course they have taken from the list of courses.

Step 2 : Pass Each review into the review classifier and generate a score . Keep the generated score as value of the review .

Step 3 : Covert any Text Field except for reviews into a numerical field by giving each entry a Specific numerical value . Eg

Step 4: For each user Calculate a score using The Score Calculation Formula Definition

A). Year : The year of study of the user among (1,2,3,4)

B). Difficulty Level : The difficulty of the course required (Beginner, Intermediate ,Advanced)

C) Length Of Course (in Months) : The time duration of the course needed (1,2,3,4,5)

D) Certification : If certificate is required (Yes,No )

Score = *A*/4 + *B*/3 + *C*/5 + *D*

Step 5 :Train The Linear Regression classifier for the above calculated score of each

user . With their input being the four numerical columns and the label being the score .

Step 6: For any new user get the four input fields and calculate the score using the classifier.

Step 7 : Get the users with same score as the new User and for the reviews where review Score is high recommend them as recommendation to the user.

END

4. Work Done

The sentiment analysis model proposed for this approach uses Gaussian Naive Bayes classifier which is initially a binary classifier and in the study of the proposed approach a multinomial Naive Bayes is seen to have greater impact as binary model just classifies the review as good/bad but a multinomial classifier will be able to classify the review on the scale of 1- 5 . The classifier accuracy is 70 percent based on testing data taken from the course dataset. This approach justifies the use of Naive Bayes classifier as it has been seen to have a high accuracy for text based classification and sentiment analysis [5].

For finding the similar users the defining features shown to have greater impact than other s include difficulty level of the course, length of course etc. The initial algorithm using the

mentioned features was trained on Linear Regression model . The further study based on the distance parameters led to cosine similarity with threshold of 0.

The hierarchal clustering model applied uses method as cosine and for the different values of threshold the number of clusters also varied with the optimum value being between 0.005 to 0.001 where the following table shows the comparison of threshold versus classes formed .

| Distance Metric | Parameter Name | Value |
| --- | --- | --- |
| Cosine\_Similarity | Threshold | Classes Formed |
| 0.001 | 15 |
| 0.002 | 7 |
| 0.003 | 4 |
| 0.004 | 3 |
| 0.005 | 3 |
| Jaccard Similarity Score | 0.9 | 36 |
| 0.5 | 36 |
| 0.01 | 36 |
| 0.09 | 36 |
| 0.001 | 36 |