Collaborative Filtering Based Course Workload Prediction

Yanqing Ma
University of Illinois at Urbana-Champaign
yma34@illinois.edu

Yuxi Gu
University of Illinois at Urbana-Champaign,
yuxigu2@illinois.edu

Junpeng Qiu
University of Illinois at Urbana-Champaign
jqiu6@illinois.edu

Xuanyi Zhu
University of Illinois at Urbana-Champaign
xzhu42@illinois.edu

Qingkang Zhang
University of Illinois Urbana-Champaign
qzhang72@illinois.edu

Abstract
In this paper, we leverage a collaborative filtering method to help college students at UIUC decide their schedules more wisely based on their expected course workloads. We introduce our CFCSP system: Collaborative Filtering Course Schedule Planning system, which is a terminal-based application leveraging user-based collaborative filtering algorithm in order to help students plan their course schedule semester-wise based on historical data of students' perceived workloads of courses they took before. In the later sections, we mainly discuss our plans for data collection, algorithm, and system implementation as well as our evaluation method and result.

Author Keywords
Collaborative Filtering; Course Workload Prediction; HCI;

Introduction
Problem Statement
We would like to study the performance of user-based collaborative filtering model on course workload prediction.

Motivation
Nowadays, most universities and colleges in the United States offer hundreds of various courses from different majors. A college student needs to fulfill his/her degree requirement by planning coursework and arranging

schedule every semester. A light schedule with too little workload may prevent students from learning sufficient knowledge and improving themselves. On the other hand, a tight schedule with too much workload or difficulty not only possibly causes a student to fail with low GPA, but also discourages and depresses a student. In fact, for many students, planning a course schedule is a difficult and challenging process because of three reasons: 1. Every course has different levels of workload and difficulty, which determines the amount of time and efforts needed to put in. 2. College students have to manage the balance between study and personal affairs(work, social life, etc), which makes it more difficult to plan a reasonable course schedule. 3. There are no official course planning tools available to students, such that when it comes to course planning, students can only search limited information from online forums.

Therefore, advising office in the Department of Computer Science at the University of Illinois Urbana-Champaign(UIUC) tries to alleviate this burden from students by offering advisers who could help students plan their course schedules. However, during the registration season each semester, the number of students who come to the advising office is so large that they usually have to wait in a long line for advising. In addition, advisors in CS advising office usually only provide suggestions and basic information about computer science courses, such as whether a student can fulfill a specific requirement for graduation. However, advisors usually tend to not give information and suggestions about the specific course workloads about courses since workloads vary a lot depending on the background of students. Therefore, we are motivated to implement a system that can resolve this issue and help students plan their personalized schedule based on their expected workloads. In this way, we believe students are more likely to find a study-life balance and enjoy their college life.

Related Work

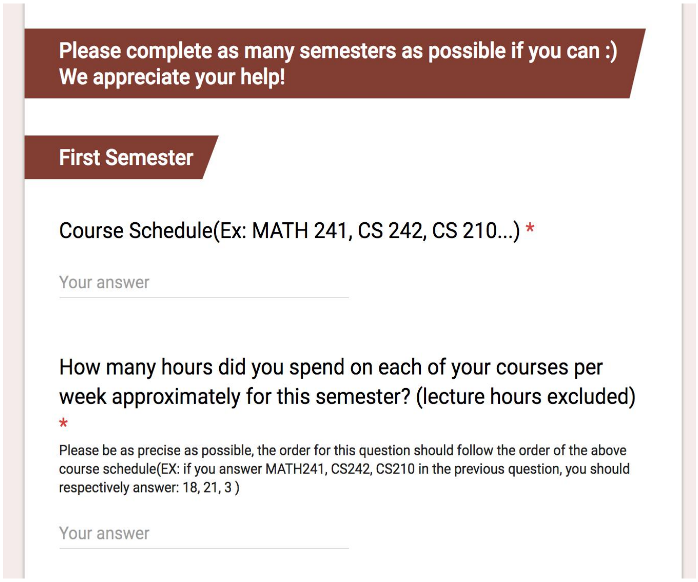
A Case Study upon Student GPA Prediction

Researchers in Information Technology University (ITU) has made an investigation into predicting student academic performance by employing three algorithms (Iqbal et al., 2017): Collaborative Filtering (CF), Matrix Factorization (MF), and Restricted Boltzmann Machines (RBM). The dataset used in the study is a list of GPA that was achieved by a particular student in a particular course in ITU (i.e., a <Student, Course, GPA> triplet). Those data are collected from 225 undergraduate students in the ECE department, so most courses on the list are related to ECE. The trained model takes a <Student, Course> tuple, and outputs the predicted GPA. However, the model studies GPA, which is the final achievement of the student. A modern understanding of academic performance shall not be based on GPA only, things like the spending time, study approach, and preparation level are as important as well. In our investigation, we predict workload (i.e., spending time per week) for students so they could schedule the semester with better time management.
Link: https://arxiv.org/pdf/1708.08744.pdf

Methodology

Data

We collected the whole data through a google form survey filled by UIUC Computer Science department students. A screenshot of our survey is shown below.
Link: https://docs.google.com/forms/d/1QT7LcY-iVZ3DPAs-bY8xBvuL4rg6CeRSklLiuPfB3tE/edit#responses


Please complete as many semesters as possible if you can :)
We appreciate your help!

First Semester

Course Schedule(Ex: MATH 241, CS 242, CS 210...) *

Your answer

How many hours did you spend on each of your courses per week approximately for this semester? (lecture hours excluded) *

Please be as precise as possible, the order for this question should follow the order of the above course schedule(EX: if you answer MATH241, CS242, CS210 in the previous question, you should respectively answer: 18, 21, 3)

Your answer

For each student who fills our survey, we ask about their course schedule each semester and corresponding workload of each course at this schedule (the workload is estimated in hours per week and lecture hours are excluded). Also, each student fills the semester-wise data for at least one semester, but up to 8 semesters. As of now, the transcript dataset contains 53 students and 220 courses, where all required courses and most 400-level elective courses in the computer science department of UIUC are included.

To evaluate the effectiveness of our CFCSP system, we conduct our empirical user study by inviting ten cs students to try our system.

The algorithm behind-- User Based Collaborative Filtering

Collaborative Filtering is an application of K-Nearest Neighbour to a recommendation dataset (user, item matrix). According to Toscher and Jahrer, collaborative filtering is the most common used ML model for predicting student grades (Toscher and Jahrer, 2010). And many education researchers employ it to identify similar students in the dataset for prediction.

	cs125	cs233	cs241	cs374 ...
student1	6 (hrs/week)	10	25	16
student2 ...	8 ...	12 ...	N/A ...	20 ...

In our study, the workload matrix is constructed as $M_{s,c}$, where c denotes the course, and s denotes the student. Hence, the row vector is a student’ s workload in corresponding courses. Note that such matrix could have unavailable entries if one student has not taken one course. The goal of CF is predicting those entries. Based on all available entries in the matrix, the model finds a list of k nearest neighbors (N) of the target student row $r_{s,..}$ using cosine similarity. Given the KNN resulting neighbors list, for an unavailable entry $r_{s,\ target}$, using user-based CF approach, the resulting numerical weekly workload hours are computed as weighted average of deviations from neighbor’ s mean and adding it to target student’ s mean rating. Deviations are used to adjust for the student associated biases. Student biases occur as certain students may tend always to report high or low hours to all courses.

$$p_{a,i} = \overline{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \overline{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}}$$

p(a, i) is the prediction for target student a for course i, w(a, u) is the similarity between students a and u, and K is the neighborhood of most similar students.

Systems

We are building a terminal user interface which takes a student’s email, completed courses and respective workload as input and outputs the predicted course workload for the course which the student asks.

```
Type in your email
for example, 'tracy@gmail.com'
>>yma34@illinois.edu

Type in your taken courses with corresponding workload sperated by spaces
for example, 'cs125:10 cs233:11 cs225:16 cs241:25'
>>cs125:10 cs233:8 cs225:20 cs241:25
(51, 220)

Type in course you plan to take
for example, 'cs242 cs210'
>>cs374
('yma34@illinois.edu', 50)
('cs374', 11)
(15.75, 11.507705316731125, 1.475216166015602)
prediction workload is 23.550690896583774hours
```

(a screenshot of our terminal user interface)

As shown above, when a user starts to use our system, he will be asked to type his email address, and his completed course names and each course’s corresponding workload per week in hours. For example, a user has taken CS125, CS233, CS225, CS241 before and he spent 10, 8, 20, 25 hours on the four courses respectively outside of classes. Also, he reports these four completed courses as well as the corresponding workload in our system. Then our backend algorithm will start running, where the newly added user inputs are included. And the backend algorithm is based on user-based collaborative filtering, which is a method of making automatic predictions about the interests of a user by collecting preferences information from many users. In our case, we collect other users’ perceived workloads of the courses they took before, and we predict workloads for the courses that the current user want to take. More details about collaborative filtering and how it works has been described in the methodology section.

Then, our system will prompt the user to type in the course which he would like to predict. In this case, the user asks the workload of CS374. Our collaborative filtering model returns 23.55 hours per week for this specific user. And our system can also predict the workload for a semester-wise schedule.

Intended Users

As of now, due to the nature of our collected data, our intended users are all current and potential UIUC computer science department students. However, we designed our system in an extendable and compatible way such that it can be generalized to service all major students in a specific university when the collected data is massive enough(e.g., granted by the university official).

Evaluation And Metrics

To evaluate whether our collaborative filtering-based course workload prediction system satisfies user expectations or not, we conducted an empirical user study involving ten students who help evaluate our system. All of the interviewees are Computer Science students at the University of Illinois at Urbana-Champaign.

User Study

We asked ten of our friends and classmates to participate in our evaluation. We interviewed these ten users by either talking to them directly or sending our system to them. The user study involves two parts:

A. Predicting a user’s future course workload based on his previous courses
B. Collecting ten previous courses and workload pair from each user and apply RMSE (root mean squared error) as an evaluation metric

PART A:

We asked each of ten interviewees to provide his/her previous courses taken at UIUC and weekly hours spent on each course, respectively. There was no restriction on the course category, but each interviewee must provide at least four previous courses, which was the usual number of courses taken in one semester. (Interviewees who used our system by themselves directly typed in their input using terminal user interface. For other interviewees, we talked to them and typed in their course workload for them).

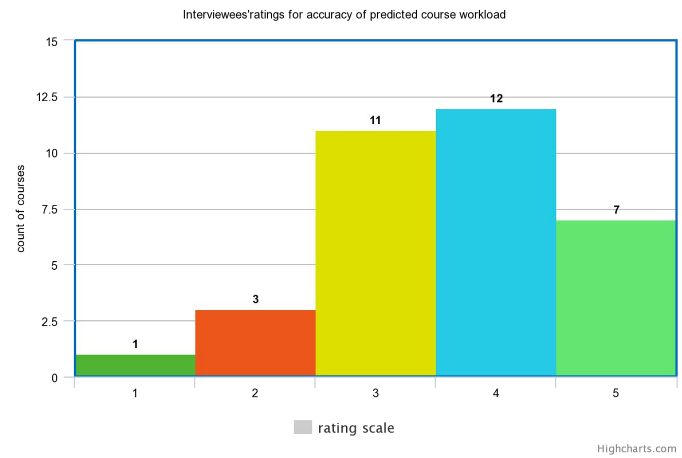
After typing in user input into the system, we asked each interviewee to provide us the courses which they would like to predict (There was no restriction on courses, but interviewees were suggested to query about CS/Engineering courses). Then, for each interviewee, we inputted courses which they queried about, and our system generated predicted workload in hours for each course.

Finally, we asked each interviewee to evaluate the effectiveness of our course workload prediction system by rating each predicted course workload accuracy compared to their expectation in a scale from one to five.

Below is a summary of courses queried by ten interviewees and ratings given by them.

User	Predicted courses	rating for each course
1	CS225, CS374, PHYS211	4, 4, 3
2	GEOG101, ECON103, CS125, STAT400, NPRE100	3,3,2,4,2
3	PHYS212, PHYS213	5,4
4	CS461, CS357, CS374	3,5,5
5	CS425, CS421, CS498WEB, HORT107	4,3,4,1
6	CHLH101, CS225, CS233, CS241	2,5,3,3
7	ECE220, ECE385	4,4
8	MATH286, PHYS213, CS473, CHEM102, PHYS214, CS446	3,4,4,3,5,3
9	CS410, CS411	4,5
10	CS241, CS126, CS427	3,5,4

As a result, we drew a histogram showing the distribution of interviewees' ratings:



The x-axis is the rating scale from 1 to 5 and y-axis is the count of predicted courses.
From the histogram, we can see the average accuracy/satisfaction of predicted course workload is approximately 3.5 out of 5 points. Also, during this user study, we have found that the accuracy of predicted course workload is directly proportional to the number of previously taken courses provided by users. In other words, as a user provides more past courses and corresponding workload as input, course workload predicted by our system for this user will be more accurate. A Second finding is that predicted workload for common/popular courses is more accurate than others. For example, most interviewees rated 4 or 5 points for accuracy of popular courses like PHYS212, CS374, CS411. A possible reason might be that these courses have more reported workload counts in our collected data. Therefore, the Collaborative-filtering model can compute a more reasonable and accurate prediction.

Part B:
In this part of the evaluation, we applied RMSE (Root Mean Squared Error) with cross-validation as an evaluation metric for our Collaborative-Filtering model.

We chose RMSE because it is one of the most popular and commonly used evaluation metrics for Collaborative Filtering.

Firstly, we asked the same ten interviewees to each report 10 previous courses and corresponding weekly workload in the unit of hours. Then, we split each interviewee's 10 courses into a training set of 8 courses and a test set of 2 courses randomly. Afterward, for each interviewee, we typed in the weekly workload of 8 courses as input data, ran the collaborative filtering algorithm, and predicted course workload for the 2 courses in the test set. We recorded each interviewee's system-predicted course workload and actual workload of courses in the test set and combined all interviewees' data. Then, we conducted cross-validation for these 20 courses by comparing the prediction by our algorithm with the actual values from interviewees, using RMSE.

```
true_val = [12,6,11,3,10,6,4,8,1,18,1,3,20,16,5,8,19,2,12,9]
pred_val = [12,12,7.65,4.57,9,6.6,7.1,6.5,3.4,20.1,2.1,6.2,20.5,13,5.5,8.96,23.2,10.5,8,9.5]
rms = sqrt(mean_squared_error(true_val, pred_val))
print(rms)
```

```
3.170875273485225
```

As shown above, the RMSE of our model is 3.17. Considering the datum ranges from 0 to 30 and the property of course data, our model accuracy is at a medium level. We think the main reason is that we have a small dataset of only 53 users collected from the Google Form.

Future Work

Our system currently only uses user-based collaborative filtering for predicting unknown course workload for a specific user, based on his/her previous schedule and workload data. And one future improvement is to use item-based collaborative filtering, which is likely to have a better result when the number of users is significantly smaller compared to the number of courses. Also, we only use the cosine similarity measure in the collaborative filtering method. Future researchers could consider trying different similarity measures, such as Pearson and Euclidean, then comparing the results to discover which similarity measure generates a more reasonable result.

Also, another work could be done in the future is not only to ask people about the workload of each course but also asks them about the degree of difficulty in the data collection process. After that, researchers can build a collaborative filtering based system that helps users evaluate both the difficulties and workloads of a planned semester-wise course schedule. In this way, a user might find a better balance between workloads and difficulties when he/she is planning his/her course schedule. Similar to this idea, other features like the usefulness of the course and ratings of professors could be incorporated as well to build a better system that has more comprehensive and accurate predictions based on students' personalized needs.

References

1.https://en.wikipedia.org/wiki/Collaborative_filtering
2.IQBAL, Z., QADIR, J., MIAN, N., and KAMIRAN, F. 2017. Machine Learning Based Student Grade Prediction: a Case Study. Information Technology University.
3.TOSCHER, A. AND JAHRER, M. 2010. Collaborative filtering applied to educational data mining. KDD cup.
4.THAI-NGHE, N., DRUMOND, L., HORVATH ´ , T., KROHN-GRIMBERGHE, A., NANOPOULOS, A., AND SCHMIDT-THIEME, L. 2011. Factorization techniques for predicting student performance. Educational Recommender Systems and Technologies: Practices and Challenges, 129–153.
5.SAARELA, M. AND KARKK AINEN, T. 2015. Analysing student performance using sparse data of core bachelor courses. JEDM-Journal of Educational Data Mining 7, 1, 3–32