CS411 Project Report

Zecheng Zhang[1, +], **Jason Situ**[2, +], **Linzi Meng**[3, +], **and Yuxin Xiao**[4, +]

[1]University of Illinois at Urbana Champaign, Department of Computer Science, zzhan147@illinois.edu
[2]University of Illinois at Urbana Champaign, Department of Computer Science, junsity2@illinois.edu
[3]University of Illinois at Urbana Champaign, Department of Computer Science, lmeng10@illinois.edu
[4]University of Illinois at Urbana Champaign, Department of Computer Science, yuxinx2@illinois.edu
[+]Group: 28. Head of CS Department

ABSTRACT

The project we did aims to help CS students to make wiser choices when registering computer science classes of UIUC in general. Each registered student has a personal profile on our website showing what classes he or she can still take in the chosen computer science track. By providing the hotness and average GPA of each class, students will no longer face the problem of what classes are more popular or think which classes can give a higher GPA. In addition, we provide search bars for students to search the classes they want. Therefore, students are able to plan accordingly what classes to take in the future semesters in their own tracks with the help of our website.

Usefulness

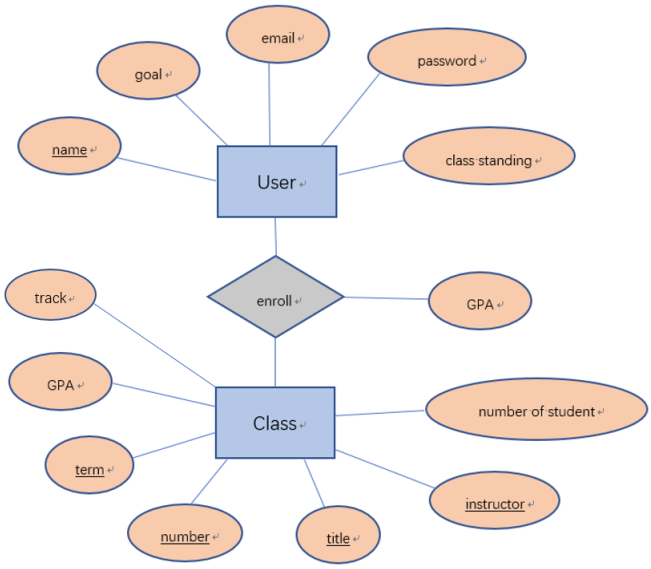
- Our project helps students to get clearer about what computer science classes they should take in order to master the track of their computer science classes.
- Our project enables students to have a more insightful ideas about what classes are more popular among fellow students and what classes more likely to give a higher GPA based on the accumulated data from previous semesters.
- The project also provides ranking clustering for professors based on the average grades they gave to students and how many students they taught.
- Students can search for computer science classes based on the average GPA showed.
- Students can see the users' average GPA for each track the users have registered on our website as well as the average GPA of all registered users.

Data

- The database "classInfo" refers to the relevant data we was given from Prof. Fagen's University of Illinois's GPA Dataset.
- The database "enroll" stores the users' own GPA records for their corresponding classes.
- The database "rightEvent" stores the message that the user puts in on the "Add New Goal" dialogue prompt.
- The database "user" holds the information that the user puts in on the sign-up page, including the user's name, email, encrypted password, current class standing, and chosen track.

- The database ”trackClass” holds the information which track the class is in.

ER Diagram and Schema



Schema:
User(name, email, password, classStanding, track)
Class(term, number, title, instructor, NumOfStudent, gpa, track)
enroll(name, term, number, title, instructor, gpa)

Data Collection

We collected the class size and average GPA of all the Computer Science courses in the past semesters from Prof. Fagen’s University of Illinois’s GPA Dataset. We rearranged the data based on the year at which the course was offered and the course number. We also used API to get real course data from the course explorer website.

Functionality

- The users can specify one Computer Science track of their choices.
- The users can leave messages as their semester goals.
- The users can change their class standings, chosen tracks and basic information in the setting.
- The users can search for specific courses on the top class search bar.
- The users can keep track of specific classs on the right bar.
- The users can view the GPA information on the relevant course page.
- Other advanced functions will be given introduction below.

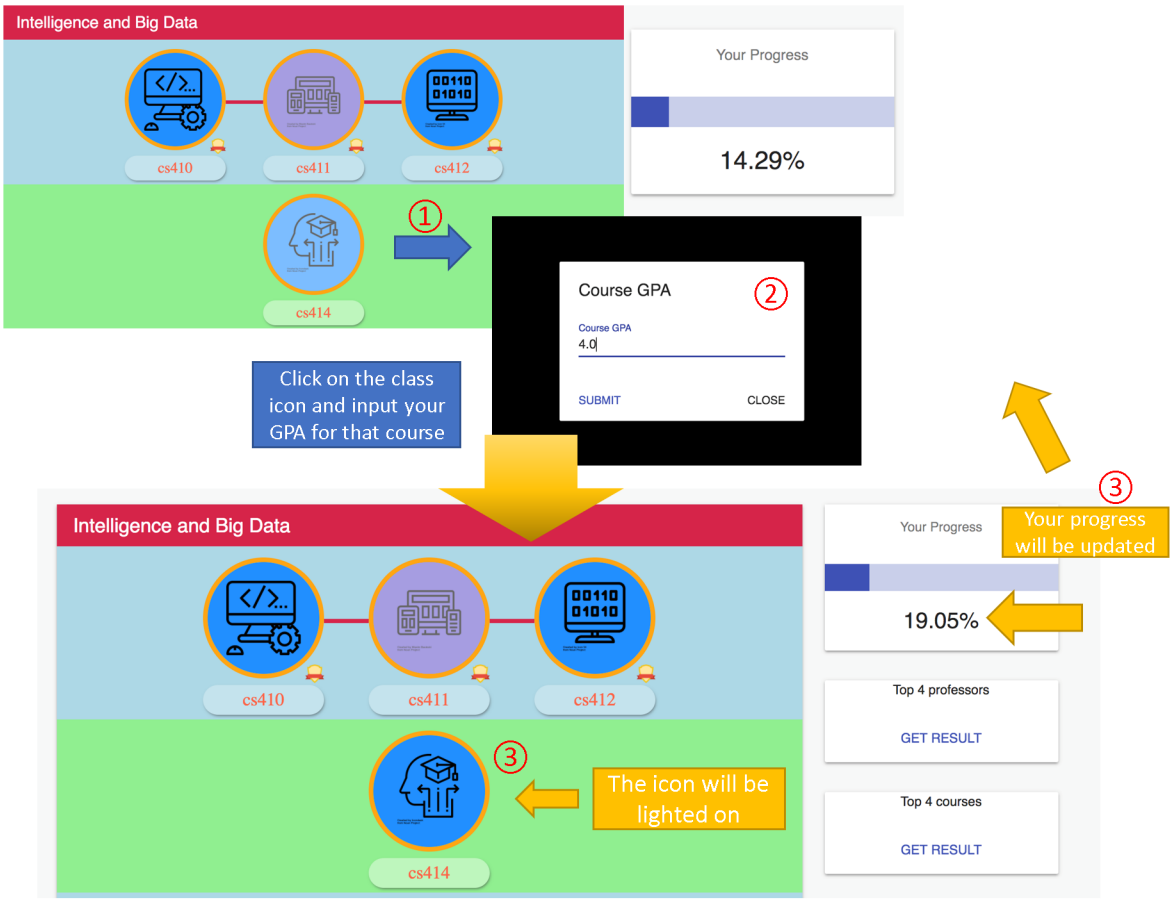
Basic Function

One basic function is to allow a user to put in his or her GPA on a specific class. This information will then be stored into the class database and the corresponding visualization on the webpage will be updated after the insertion.

SQL Code Snippet

```sql
-- BLOCK update_user_setting
UPDATE
  user
SET
  year = :u_year,
  direction = :u_direction
WHERE
  email = :u_email;

-- BLOCK getUserTakenClass
SELECT *
FROM enroll
WHERE email = :u_email

-- BLOCK addClass
INSERT INTO enroll (email, class, gpa)
VALUES (:u_email, :u_class, :u_gpa);

-- BLOCK getTrackClass
SELECT class
FROM trackClass
WHERE track = :u_track

-- BLOCK getTrackAvgGPA
SELECT t.track, AVG(e.gpa)
FROM trackClass t, enroll e
WHERE t.class = e.class
GROUP BY t.track

-- BLOCK getAllUserAvgGPA
SELECT u.email, AVG(e.gpa)
FROM user u, enroll e
WHERE u.email = e.email
GROUP BY u.email
```

Dataflow


Intelligence and Big Data
cs410
cs411
cs412
cs414
Your Progress
14.29%
1
Course GPA
2
Course GPA
4.0
SUBMIT
CLOSE
Click on the class icon and input your GPA for that course
3
Your progress will be updated
Your Progress
19.05%
Top 4 professors
GET RESULT
Top 4 courses
GET RESULT
3
The icon will be lighted on


Top 4 professors
GET RESULT
1
Top 4 courses
GET RESULT
3
After click on the GET RESULT button, return the results of each block
2
Top 4 professors
Han, Jiawei
Karahalios, Kyratso G
DeJong, Gerald F
Zhai, Chengxiang
GET RESULT
4
Top 4 courses
CS512
CS410
CS467
CS440
GET RESULT

Advanced Functions

In this project we have two main advanced functions. Following are more detailed descriptions of them.

1. Bi-typed Heterogeneous Information Network

The first advanced function we implement is a **Heterogeneous Information Network**[1]. We use this algorithm to both ranking and clustering classes and instructors according to their average GPA and number of registered students information. Different from homogeneous information network, heterogeneous information network has multiple object types and link types. As showed in figure below, left side nodes have type classes and right side nodes have type instructors. Each instructor can teach different classes and each class can be taught by different instructors. The real line in the figure exhibits these relations. In addition, if both professors have taught same classes, there is also association between them which in the figure denoted as dotted lines.

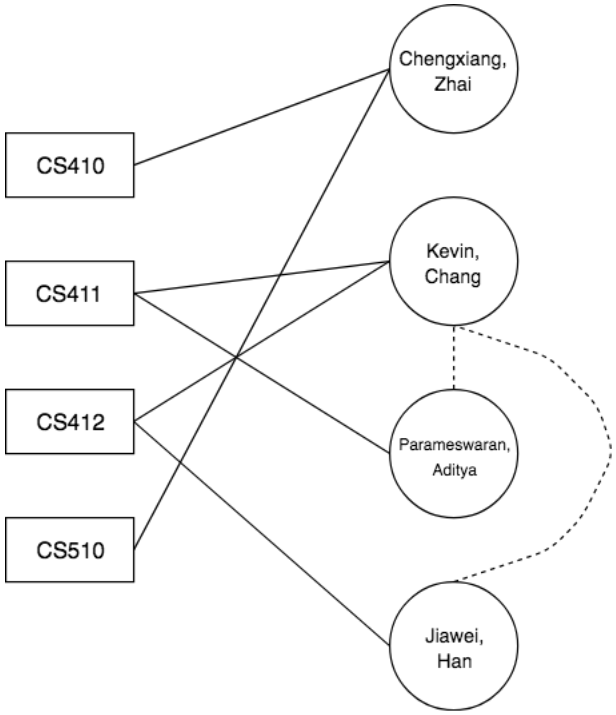


Figure 1. Heterogeneous Information Network overview in our project.

In this project and as shown in the figure, we propagate the GPA and number of registered students information among classes and instructors. During the propagation, we use algorithm for our bi-typed network model based on the heuristic from **RankClus**[2]. Let X represents classes, Y represents instructors and W is representation of information network, we can derive following rules:

- Rule 1. Highly ranked instructors usually teach in highly ranked classes:

$$\overrightarrow{R_Y}(j) = \sum_{i=1}^{m} W_{XY}\overrightarrow{R_X}(i)$$

- Rule 2. Highly ranked classes usually taught by highly ranked instructors:

$$\overrightarrow{R_X}(i) = \sum_{j=1}^{n} W_{YX}\overrightarrow{R_Y}(j)$$

- Rule 3. Highly ranked instructors usually will be enhanced a bit if other highly ranked instructors also have taught same classes:

$$\overrightarrow{R_Y}(i) = \alpha \times \sum_{j=1}^{m} W_{XY}\overrightarrow{R_X}(j) + (1-\alpha) \times \sum_{j=1}^{n} W_{YY}\overrightarrow{R_Y}(j)$$

Using these rules we implement algorithms that can iteratively and mutually ranking and clustering classes and instructors based on information of average GPA and number of registered students for each class of each instructor. After about several iterations (heterogeneous information network usually require relatively small number of steps for the size of data in our project), we can easily see that professor Chengxiang Zhai is ranked very high. This matches the fact that professor Zhai usually teaches CS410 and CS510 and in both classes he give almost the most number of As compared to other CS classes. Also, from the class ranking we can also see that CS410 and CS510 are also ranked very high in classes which matches exactly our heuristics and rules.

We consider this function as an advanced function for following reasons. First, heterogeneous information network is a kind of relative new techniques which there almost exists no corresponding packages to learn. Second, for computer science classes, data size is not too large, so it is not very reasonable to use deep learning related methods in this situation. And compared to normal clustering, ranking or generative model, heterogeneous information network can provide better result since it includes different type information and linkage information. Third, heterogeneous information network in this size of data is efficient, which can return both the ranking and clustering in one pass.

2. Contiguous Frequent Pattern Mining

In our project we also implement a contiguous frequent pattern mining algorithm that can help to mine frequent patterns of classes in number of registered students and average GPAs. More details can be shown by the figure below:

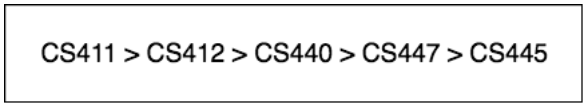


Figure 2. Frequent Pattern Mined for Number of Registrations for Data and Intelligence Track (Threshold 3 semesters)

Figure above shows that there is such pattern for number of registered students for those classes and this pattern appears at least 3 semesters. In order to visualize well, we only choose part of GPA pattern and number of registered students patterns on our project. The choice of only implementing the contiguous one also depends on the visualizing requirements.

We consider this function as an advanced function for following reasons. First, it is one of the most useful pattern mining techniques[3] in the realm of data mining. Second, it is relatively hard to implement and there is no packages which can be used directly or to learn about. Third, the mining result can offer students more clear class registration and GPA pattern intuitions. Last of all, the method provide deterministic result (by almost exhaustive search) which will not influenced by other factors (such as some machine learning models can be relatively unstable since undecidability in training steps).

Technical Challenges

- **Front End:** We use the React and Material UI for front end. Even if there are more and more people using React, but the amount of React coding, front end designing are relatively small which made us spending more time on the front end coding.

- **Back End:** We use the Express as our backend choice. Compared to PHP the coding might be more clear but it requires to install corresponding packages which sometimes will cause confusion and other complexity.

- **Advanced function:** The advanced functions we choose are both relatively novel ones and there are almost no existing packages in Python.

- **Cpanel:** One of the problem of provided Cpanel is that the disk is very limited and the Python version is 2.7. After installing the virtual environment, Node.js packages and other Python packages the disk will not have much remain. So, we tried our best to not use the install required Python packages in our advanced function which help us save some memory usages.

Changes

Initially we want to use CSS, HTML, PHP and JavaScript for the front end and back end. But this setting seems to be a little bit chaos, and it will be relatively hard to use PHP and CSS. So we use the SCSS instead and use the Express.js as backend. Also, we want to use some clustering methods on clustering the high GPA classes and professors. However, we found that the amount of data in this situation can't train a good network such as Variational Autoencoder. Thus, we change to use a novel approach, heterogeneous information network, which can rank and cluster together and require somehow less data in this case. And for another advanced function we want to make a "User GPA Prediction" function first, but we later found that that was not possible. GPA relies on not only classes and professors but also depend on users' own study and other things. Thus, we change our advanced function into a more "deterministic" one which can provide interesting patterns of computer science classes.

Contributions

We managed our teamwork by setting deadlines for subtasks and having group meetings on a regular basis. Following is the distribution of contributions among our group:

- **Zecheng Zhang:** Part of front-end, back-end and SQL. Advanced functions and part of report.

- **Jason Situ:** Front-End, set-up the hosting server, database connection with SQL parsing. Create API for front and back-end communication.

- **Linzi Meng:** ER diagram and relational schema. Video demo and part of SQL and report.

- **Yuxin Xiao:** Part of SQL and report.

References

1. Sun, Y. & Han, J. Mining heterogeneous information networks: principles and methodologies. *Synth. Lect. on Data Min. Knowl. Discov.* 1–2 (2012).

2. Sun, Y. Rankclus: integrating clustering with ranking for heterogeneous information network analysis. *In Proc. 12th Int. Conf. on Extending Database Technol. Adv. Database Technol. ACM* 565–576 (2009).

3. Aggarwal, C. C. & Han, J. Frequent pattern mining. *Springer* (2014).