A Personalized Course Recommender System for Undergraduate Students

Thi-Hai-Yen Vuong
School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan
Email: yenvth@jaist.ac.jp

Thi-Thu Trinh, Hong-Duyen Ha, and Xuan-Hieu Phan
University of Engineering and Technology, Vietnam National University Hanoi, Vietnam
Email: {thutt_57, 14020066, hieupx}@vnu.edu.vn

Thi-Oanh Tran
International School, Vietnam National University Hanoi, Vietnam
Email: oanhtt@isvnu.vn

***Abstract*— In this paper, we describe a web-based course recommender system which has a variety of functions for supporting students in academic fields. The system's goal is to help students manage their own study progress in a more effective way. With this system, students would achieve better performance, find it easy to select courses without wondering which will give them high score or which are better for their future job or career path. The core functionalities of the system are partly based on recommendation methods. The first task is the prediction of student performance to help them estimate their performances on selected courses. These predicted results help students choose courses that are appropriate for them to get higher scores. Second, the system will automatically build a study strategy for next two years or even entirely four academic years based on their ability and related information. Our system also provides students a general view of their learning status: profile, grades, taken courses, credits, important school news, time table, etc. By using these functionalities, students will have a guidance and an up-to-date studying path to follow, as well as efficiently complete their school work.**

***Index Terms*—student performance prediction, recommender system, study strategy**

I. Introduction

Today, more and more universities are using credit systems in their education and training activity. Academic credit systems assess students' progress in their studies. Students are required to earn a certain number of credits in order to be entitled to full-time student status. Each course is worth a certain number of credit points determined by different criteria including student's workload, learning outcome, etc. In Vietnam, students can gain credits by successfully completing their selected courses. Hence, choosing the right courses is an important decision as it can influence the students' future success. And for those who are doing a course they are not happy with, they will eventually study badly. Unfortunately, students are usually confused when choosing these elective courses because they do not know which ones are most suitable for them. There are a couple of reasons for this:

- Students do not usually know the background knowledge (required skills, how the course fit their studying preference) about these elective courses.
- Students do not know how useful the course is to support their major and career path after graduation.
- Students may not choose the right number of courses in order to meet hard constraints from universities such as the minimum or the maximum number of credits required in each field/ area in a term, etc.

Thus, the current solution is to select based on the manual guidance from their tutors/teachers. This process is really expensive, especially in some situations where the tutors/teachers do not know much about the background knowledge as well as understand the ability of the students. In order to overcome these problems, we have proposed a Personalization and Recommendation System for Education (PRE).

In our system, we proposed the use of recommender system techniques such as collaborative filtering (CF), content-based recommendation, as well as several other techniques such as min-cost max-flow algorithms, latent Dirichlet allocation (LDA) in order to:

- Successfully match students with courses in each semester as well as in their study path in four academic years that are suited to both their ability/preferences and goals. In addition, courses recommended by the system not only satisfy these soft constraints but also meet hard constraints of the university such as total number of credits, number of credits for each area, pre-requisite courses or co-requisite courses.

Manuscript received December 20, 2018; revised March 21, 2019.

doi: 10.18178/ijlt.5.3.181-190

- Predict the performance of students on unlearned courses, so that the students may know, at least, some information about their (predicted) performance on those courses. This information can also be used to guide the course recommender in order to suggest the courses that students may earn high scores if they notice that it is more important criterion.
- Provide personal information of the student, visualize results, reports, statistics and analysis about the current academic status of students in an easy-to-use and friendly way.
- Integrate a variety of reports that help optimize course schedules and also notify warning to help university offer support to students with low performance.

Academically, in education and training, recommender system is used to recommend courses to students. At a basic level, a recommender system just need to know students major to suggest them most related courses. Nonetheless, this kind of recommender is not personalized. Moreover, students also have to follow university requirements so that finally they can get diploma of bachelor. As a result, a course recommendation which combines traditional score strategies and constraints could be the best to give personalized recommendation with existed constraints satisfied. There are several successful course recommender systems as Course Rank at Stanford University of Parameswaran *et al*. [1], degree compass from Austin Peay State University of Toscher et al. [2] or student performance prediction and course recommendation system of Thai-Nghe *et al*. in CanTho University [3].

II. Related Work

In our work, we will present our effort to give suggestion and advice to students during their learning process. Our system has two main types of recommendation: (1) student performance prediction and (2) study-path recommendation. There have been several studies working on course recommendation and some have been successfully applied in real life so far. Traditional methods are often used in Recommendation System (RS) such as association rules [4], [5], collaborative filtering [6], [7], content base [8], [9], and so on.

In course recommendation task, one of the most famous methods used in RS is the association rules. The most relevant work is RARE [10], where a framework was proposed to recommend course based on association rules. They incorporated association rules derived from alumni's data along with student ratings for recommendation. Other related work, they constructed course trees [11], upgrading them by using the association rules. These rules were created from searching for relationships among subjects by mining data about the course conducted by students, then comparing them with the current course tree to repeat upgrading it. However, these methods are not suitable to meet the complex constraints in our recommendation problem.

One of the most relevant works to our problem is Course Rank [1] where the recommendation course supports the student learning plan. Course rank will "take from", is a positive number, is a set of given courses. Each course will be associated with a score that represents the importance of the course. The score of each course for each student is different depending on the features of the student. Similar to our work, the system finds a minimum set of courses that satisfy set of requirements and constraints with the maximum score that shows the most useful for their personal concern. The extension in Course Rank [1] refers to meeting other various conditions using some of the approximation algorithms. As in our work, prerequisite constraints are unnecessary, study-path recommendation consists of two steps: (1) create a list (without arrangement) of courses using course recommendation with dynamic score and (2) use an ordering algorithm to determine the location of the courses in the study-path.

In the student performance prediction (SPP) task, there are many prediction models [12] which can be categorized into two main approaches below. In the first approach, authors usually formulate it as a classification or regression problem and use some typical machine learning algorithms such as SVM [13]-[15], linear regression [16], [17], decision tree [18], [19], ANN [20]-[22], etc. to build and test models at both course and degree levels. In the second approach, the PSP task can be seen as a rating prediction problem in recommender systems [2]-[3]. The authors realized a similarity between the PSP task and the rating prediction problem where students, courses, and marks can be mapped as users, items, and rating values, respectively. Once mapped, we can apply any collaborative filtering techniques to build prediction models.

Most previous work focuses on e-learning, not many studies were dedicated to academic systems. Moreover, nowadays when academic credit systems are widely used in universities/colleges, the problem of predicting student performance is becoming more and more challenging. Therefore, in this work, we will concentrate on PSP at the course level in academic systems with some changes. We target our system at predicting students' marks in order to help them know, at least, some information about their (predicted) performance on the courses, and may determine which ones are appropriate for their background and ability. About the features, we propose an additional feature set based on courses-related skills to improve the performance of regression–based prediction models effectively. In addition, to utilize effectively the outputs of these two approaches, we will also propose a simple hybrid method using the linear combination to enhance the performance of the final prediction system.

III. System Design of Personalization and Recommendation System for Education

In this paper, we describe a web-based system which has various functions to support students for many academic purposes. The system's mission is to help students manage their own study process in a more effective way. Fig. 1 shows the system architecture.

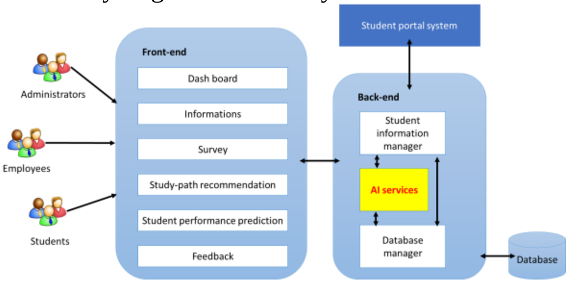

Figure 1. The overall architecture of PRE system.

Users communicate with the system through the front-end:

- Dashboard: this module is to provide statistics and reports about students' current academic status to give them an overview of their academic progress. Information such as the history of the learning process, student average scores are analyzed and presented in the tabular and graphical form. A warning message will be given if a student has poor academic performance and the system will provide some tips to improve the learning quality.
- Information: this module displays student information such as full name, student ID, birthday, class, scores, and extended students' information, etc.
- Survey: help the system to give suggestions, student's extended information should be provided as fully as possible. To collect this information, we created a survey that included a list of questions about their interests, abilities and career orientation. For the first system log-in, they will be asked to complete this survey.
- Study-path recommendation: the system will provide students with a study-path based on the student's background information, the learning process information and the extended information. Study-path includes a list of subjects and they are grouped according to each semester.
- Student performance prediction: This module gives a prediction of mark for student's unlearn subjects.
- Feedback: this is one way of evaluating the effectiveness of the system, study-path recommendation, student performance prediction. In addition, we can upgrade and improve the system from the user's comments.

When a user logs in the system, the front-end will connect to the back-end and retrieve information to display. The back-end consists of three modules:

- Student information manager: manages and automatically collects basic and extended student information recommender system. Student's basic information in our work is their names, student ID, classes, study results, etc. This information is assembled from the third-party system and student portal system.
- AI service: In this module, the system will give suggestions related to the learning process such as study-path recommendation, semester course recommendation, and student performance prediction.
- Database manager: the connection, interaction with the database will be managed by this module.

IV. The AI Service

A. *Student Performance Prediction*

If we can predict the performance of student on unlearned courses, students may know at least some information about their performance on these courses, therefore they will make a more reasonable decision in choosing one course that it might be suitable to his/her ability.

The solution to this problem came when we realized there is a mapping between the task of predicting student performance and the rating prediction task in recommender systems where students, courses, and performances are users, items, and ratings respectively. Hence, we can use data mining and machine learning techniques to solve this problem. In our case, we applied two approaches for the solution: regression and collaborative filtering. In our work, we have used personal information (e.g., interest, skills, socio-demographics) and additional useful information about students' academic performance to effectively predict their performance. We propose a method of setting relations between courses which are based on courses' attributes [23]. This information will be used as features to build our regression–based predictors. We performed a lot of experiments and analyses for finding feature set of the predicting student performance task. We collected available information of students including gender, total cumulative GPA, GPA of previous semesters, average scores of pre-requisites courses, the semester that courses were taken we also investigate another type of attributes that might affect predicting results.

B. *Study-path Recommendation*

Most students, especially early-stage students, do not have an overview of their specific learning plan. If we can provide a suitable study-path to students' abilities and interests, students may have a clearer learning orientation to have better quality.

As mentioned before, the purpose of this paper is building a study strategy for entire 4 years by considering multiple constraints so that students at our university not only can graduate, but also gain much knowledge from studying favorite courses. This is a pretty new and tough, but potential task. If these constraints are solved properly, the recommender would be a truly powerful assistant for students' study strategy recommendation. This task of course recommendation in a straightforward performance just suggests favorite courses to students without taking into account constraints from university or from student's interest as well. In each university, the educational framework is different, thus constraints so that students can graduate is also different. Moreover, to give personalized recommendation to students, apart from students' information as an academic transcript, students' interests are expected to be the constraints giving the individualized recommendation with high satisfaction. In our work, we grouped these constraints into hard constraints and soft constraints.

- Hard constraints are constraints so that students can graduate, include:
    - Credit constraint: each course has its own number of credit. When choosing courses, students have to check if the number of credit is enough as required or not.
    - Prerequisite constraint: to be able to study some courses, students have to finish other courses which are called as prerequisite courses.
- Soft constraints are constraints surveyed about their personal information:
    - Oriented career: It can be said that not every student can determine their careers even for seniors. However, the final target of any studying is getting a job. To serve students who already have oriented career, this constraint was added to the paper.
    - Interest-based: We conduct the result by surveying students, asking them to give star rating for their interests in 4 skills as math, English, programing and learning by Heart. Matching these skills with suitable courses to recommend to student.

In short, multiple constraints are considered based on graduation constraints from our university and real existed interest constraints among students. This paper not only purely solves the problem of course recommendation using traditional score strategies, but also deals with the matter of multiple constraints as well. In this paper, we propose a model using min-cost max-flow algorithm [9], [24]-[25] to solve both constraints. Below we will step by step solve each constraint, finally recommend to student the most desirable list of courses.

1) Credit constraint-based recommendation

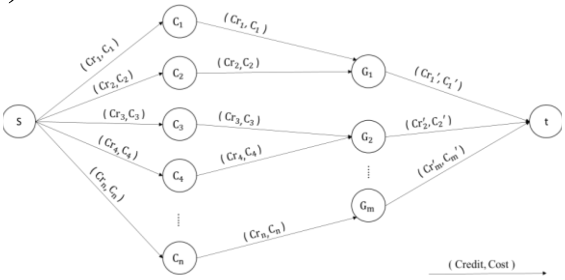

Figure 2. Credit constraint under min-cost max-flow algorithm.

When applying credit education system, students are required to study at least enough the given number of credits. In our educational system, each course is belonged to a certain group as general knowledge group, major related group and so on. To graduate successfully, students have to choose courses so that the total number of credits in each group is satisfied. In each group of courses, students just need to choose enough courses to satisfy the credit constraints. In our work, we try to recommend to students the at least courses so that student can save time as much as possible. Apply max-flow algorithm for the credit constraint, another corresponding graph can be built as Fig. 2 followed.

The source node *s* denotes for student and the target node *t* is considered as the final target. $(c_1, c_2, ..., c_n)$ set of optional courses, denoted as set **A** $(G_1, G_2, ..., G_n)$is set of course groups, denoted as set **B**. An edge (cr_i, c_i)denotes for *(credit, cost)* with meaning that if taking course *i*, student can gain a number of credit as cr_i and usefulness of course *i* is equal to $Course_1$as c_i was denoted for cost. Cost is converted from *1 – score*, here score is the usefulness value of course. Under the min-cost max-flow problem views, the above-directed graph will be created as:

- The capacity edge from *s* to *c* is equal to the credit of course *c.*
- Each course in **A** belongs to one group in **B** was created a directed edge with the capacity is also equal to the credit of course *c.*
- For each group of courses in **B**, the maximum capacity is set as the at least number of credit required in each group.

The max-flow algorithm is applied in this graph to make sure that the total number of credit at the final node is satisfied. Moreover, at each small group in B, the exiting credits is always equal to the entering credits. In short, with the support of max-flow algorithm, credit constraint can be checked easily, moreover it also helps the recommender suggest the at least number of courses to save time for students. The next part will explain in detail the role of cost or score in the min-cost max-flow graph.

2) The usefulness evaluation of courses

Among many sets of courses which satisfy hard constraints, determining which set is the best suitable for specific student can be said as the most important step of course recommendation. For this reason, it's essential to evaluate the usefulness of the courses called as score in the recommender. Hence, the higher *score* the more help of the course that students can be received. With $cost = 1 - score$ the problem of course recommendation can be considered as the min-cost max-flow problem [24]-[25] - the extension of max-flow problem. It means that the recommended courses satisfy not only the credit constraint but also the other soft constraints which are integrated in score. The score can be considered as the most significant factor which affects the quality of course recommendation. There are many factors affecting to scores, including records of previous students, popularity, ratings, student's preference and so on. The rest of this part will focus on how to make the best score to integrate into the course recommendation model. In the scope of this paper, the score will be calculated following the description at Fig. 3.

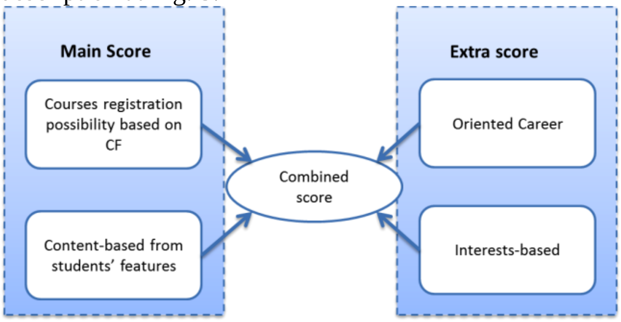

Figure 3. Methods to calculate score's value.

The first method: Course registration possibility based on CF

Academic records from 721 graduated students are used as the training data. When a new student takes part in course recommendation, information of their completed courses will be used in the collaborative filtering. In details, if the course was already finished, its rating would be equal to 1. Vice versa, if the course was not completed, its rating would be equal to 0. These students' information will be used in collaborative filtering by matrix factorization method. The output of this technique gives us the scores which are in interval of 0 and 1, named as $score_1$.

These scores predict the course enrollment possibility of courses which have not learned by students. Two tables below contain Table I and Table II which show respectively the input and the output of the matrix Factorization technique in collaborative filtering.

TABLE I. INPUT OF COURSE ENROLLMENT POSSIBILITY PREDICTION.

No.	StudentID	Course_1	Course_2	…	Course_n
1	11020711	1	0	…	?
2	11020160	0	?	…	1
3	11020117	?	0	…	0
…	…	…	…	…	…

TABLE II. SAMPLE OF OUTPUT OF COURSE ENROLLMENT POSSIBILITY.

No.	StudentID	Course_1	Course_2	…	Course_n
1	11020711	1	0	…	0.0023
2	11020160	0	0.917	…	1
3	11020117	0.082	0	…	0
…	…	…	…	…	…

The second method: Student performance prediction

This method uses the results of student performance prediction task above. Output of this method is predicted score for courses that student has not learned yet, The scores used in this paper will be converted into the range of 0 and 1 by dividing by 10. Finally, this score is denoted as $score_2$.

The third method: Oriented Career-based

Career, for almost all students, is the final target of education. Courses taught at university are more or less related to occupation after graduation. It's necessary to survey experts' opinions relevant level between courses and practical careers based on star rating from 1 to 5. Table III shows some examples of survey from experts about relative levels between courses and oriented careers.

TABLE III. EXAMPLE OF RELATIVE LEVELS BETWEEN COURSES AND CAREERS

Careers	Software Engineer	Mobile Developer	Web Developer	…	IT security
Course_1	4	3	4	…	4
Course_2	4	4	3	…	2
…	…	…	…	…	…
Course_n	3	4	3	…	3

By doing this way, each course was given a number as its relevant level to a certain career. This method can be considered as be the third way to create score. The score in interval of [0, 1], $score_3$ is converted by multiple with 0.2.

The fourth method: Interest-based

Consider Math, Programming, English, Learning by heart as 4 essential skills to learn courses in Information Technology department. Students will give their interested level for each skill respectively in range 1 to 5 in form of vector (s_1, s_2, s_3, s_4). This vector will be compared its similarity with another vector from experts' evaluation of skills needed for each course by using cosine similarity between these two vectors as shown in Equation 1:

$$\text{Similarity} = \cos(\theta) = \frac{A.B}{||A|.|B||} = \frac{\sum_{i=1}^{m} A_i.B_i}{\sqrt{\sum_{i=1}^{m} A_i^2}\sqrt{\sum_{i=1}^{m} B_i^2}} \quad (1)$$

m is the number of vector dimensions or skills, contemporarily equal to 4.

Combination of main scores:

Main scores can be considered as the "kernel" of course evaluation. We could not choose randomly only one of main scores to make recommendation, but need to combine them into one score. Therefore, one of the methods we chose to combine $score_1$ and $score_2$ is using its product as shown in Equation 2 below.

$$score_5 = score_1 \times score_2 \quad (2)$$

Combination of all scores:

Scores based on student-oriented careers and interests are two added scores we proposed to help recommender satisfy various soft constraints from clients. In our model, we need only one value of score; therefore, whenever there is another method to create score, we have to combine all of scores into one. Linear combination gives us an acceptable formula as shown in Equation 3:

$$CombinedScore = \sum_{i=3}^{n} \alpha_i score_i \quad (3)$$

n is the number of methods above to calculate score, $n = 5$ and $\sum_{i=3}^{n} \alpha_i = 1$.

α_5 is coefficient of combined main scores; α_3, α_4 are scores based on oriented career and interests respectively. Depending on school-year, we set the value of vector α differently.

3) Course arrangement

After using min-cost max-flow, we obtain the list of courses $C = \{c_1, c_2, ..., c_k\}$, which is unsorted. In order to build study-path, we proposed a directed graph method. The main idea is that we construct a directed graph that illustrates the relations of the prerequisite disciplines, where each vertex is a subject and each directed arc represents a prerequisite relation, $c_1 \to c_2$: c_1 is a prerequisite of c_2 and set of prerequisites $P = \langle (c_i, c_j), ..., (c_x, c_y) \rangle$. For each semester, a subset of courses (vertices) with the highest priority will be selected from the directed graph. These courses are the roots of a path in graphs, the higher the complexity of the path, the higher the priority of the course. See Algorithm 1 and Algorithm 2 for more details.

Algorithm 1 Course graph creating

```
procedure CreateCourseGraph(C, P)
    create a directed graph: G = (V, A)
    for each c ∈ C do
        v ← c
    end for
    for each p ∈ P do
        a ← p
    end for
    return the directed graph G = (V, A)
end procedure
```

Algorithm 2 Sorted list of courses generation

```
procedure GeneratingSortedListOfCourses(numSem, maxCredit, C, P)
    create a semester list: semList = ∅
    create a directed graph: SG = CreateCourseGraph(C, P)
    for 1 to numSem do
        create a list of course in a semester: courseList = ∅
        create check = FALSE
        while check = FALSE and SG is not empty do
            get v in SG where maxIn(v), maxOut(v)
                and numOfPrerequisite(v) are biggest
            if sum(credit if courseList + v) > maxCredit then
                check = True
            else
                add v to courseList
                remove v in SG
            end if
        end whileadd courseList to semList
    end for
    for each p ∈ P do
        a ← p
    end for
    return semList
end procedure
```

After the list of courses is built into a directed graph, the priority of each course should be determined to create the appropriate learning plan. This priority depends on four main factors. The first is maxIn(), which is the length of the longest directory path with current node as the tail, "maxIn()" = 0 means that the course is root of the path and no prerequisite. The second is maxOut(), which is the length of the longest directed path with current node as the head. The higher course' MaxOut is the higher priority in the studying order because this course is required for more courses. Lastly, to study a new course, prerequisites are required to be taken before. In other side, if you take a course then some it's children course would available for study. For a simple example in Fig. 4, course 4 has returned value of maxOut() is 2, maxIn() is 2.

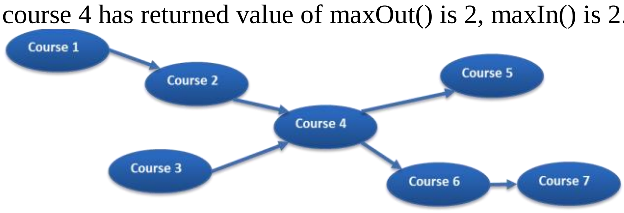

Figure 4. A simple view example for courses relationship graph.

4) Experiments.

Data Set

Offline data, with the support of the Student and Academic Affairs in our university, training data set was collected from 965 students who have graduated from Information Technology major and 124 students who have graduated in computer science from 2007 to 2011. However, the educational curriculum is quite different from year to year. To give recommendation to recent students, the curriculum of 2015 is considered as the standard educational frame in this course recommender. Therefore, all training data from graduated students from 2007 to 2011 had to be normalized following this curriculum. Finally, when all courses were uniformed based on 2015 curriculum, they were used to collaborative filtering and content-based filtering.

Data was also collected from experts' opinions by asking them to rate how courses affect the oriented careers, and which level of 4 skills: (Math, English, Programing, Learning by heart)

In terms of Online data, because the target of this paper is recommending to a student a study-path throughout 4 years at university, data of all courses finished learning by students is necessary to be later on able to evaluate the precision of course recommendation. Therefore, until now online data has been mainly surveyed from fourth year students in terms of their oriented career and their interests. Nonetheless, academic transcript among students is private, so it is difficult to ask them reveal this information. Moreover, because the training data just includes students at major Information Technology, the tested students are required from that major. There are 33 information technology students and 25 computer science students were willing to try this recommendation by showing their academic record, doing the small survey in Fig. 5 below:

Figure 5. Star rating on student's interests.

Experimental Setup

As the combination of scores equation needs to estimate the value of score, with data from students, we tested 3 times. The first time, supposed that these students just finished the first year and need to recommend courses from the second year. Similarly, when students were supposed to finish the second year, the task of course recommendation will give them recommended courses from the third year. The finally is recommending courses that third year student should learn at their fourth year at university. As a result, we find some best sets of α for the first, the second and the third year are respectively equal (1, 0, 0), (0.4, 0.2, 0.4), (0.7, 0.1, 0.2).

After setting up the value of α, by running the recommendation model, the student will be given a set of recommended optional courses which later will be compared with the real chosen course set of students to evaluate the precision of the course recommendation model.

Simple Matching Coefficient (SMC)

This similarity comparison can be said as symmetric binary comparison; therefore, Simple Matching Coefficient (SMC) shows it suitable to compare the similarity between two lists with the formula as followed Equation 4:

$$SMC = \frac{M_{00}+M_{11}}{M_{00}+M_{01}+M_{10}+M_{11}} \tag{4}$$

where:

M_{00} is the total number that courses were not predicted and real chosen.

M_{01} is the total number that courses were not predicted, but real chosen.

M_{10} is the total number that courses were not real chosen, but predicted.

M_{11} is the total number that courses were predicted and real chosen.

SMC as its definition is used for comparing the similarity between 2 binary symmetric lists, which match this problem experiment set up of comparing the recommended courses and the real chosen courses. Therefore, SMC was chosen to be the model's efficiency evaluation in this paper. The result of the SMC evaluation method done on 58 tested students were shown in Fig. 6 and Fig. 7.

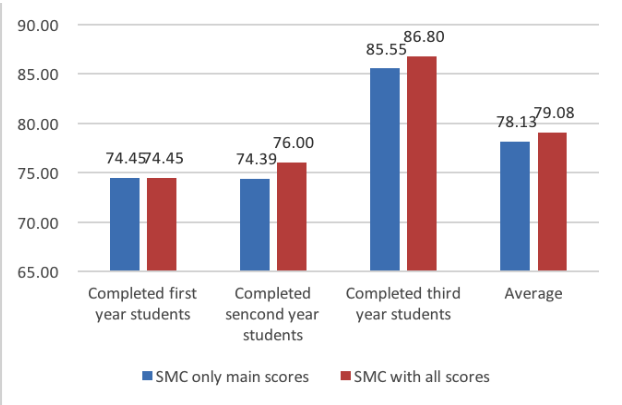

Figure 6. Testing results on 33 information technology students in 2 cases above.

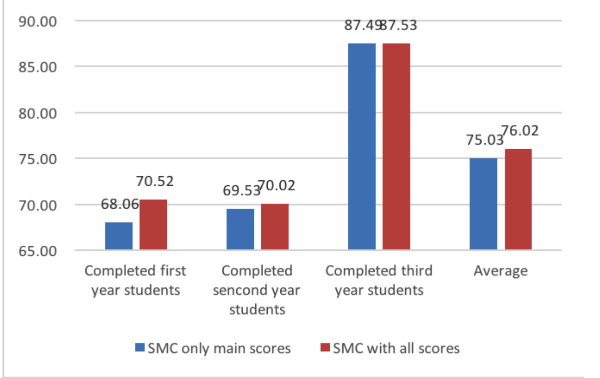

Figure 7. Testing results on 25 computer science students in 2 cases above.

This result shows that the higher-level students study, the more course recommendation can help them. The number of optional courses also affects the result a lot. With data and constraints from our university, while the total number of optional courses is about 38 courses, with the intention to give at least optional courses to students, the course recommendation model selected around 8 courses to recommend. Therefore, it requires a very exact recommendation then the precision of model can be high. As third year students, the number of recommended optional courses is getting fewer but the constraints reflect quite exactly what students tend to study. That makes the result become better.

Besides, with 58 tested students and with purposes to test the affection of adding soft constraints, only main score was integrated in the course recommendation model at first. After that, main score was integrated with extra score from soft constraints to test in second round. As the above figure showed, there is a slight change of better result. It brings belief that when being added some soft constraints, the course recommendation could be getting better.

With offline data, there are 965 students have graduated in information technology and 124 students have graduated in computer science, without soft constraints, the experiment result is described in the Table IV and Table V below:

TABLE IV. Result 965 Students have Graduated in Information Technology Without Soft Constraints.

Student	SMC
Completed first year students	0.8556
Completed second year students	0.8557
Completed third year students	0.8818
965 students have graduated in information technology	0.8556

TABLE V. Result of 124 Students Have Graduated in Computer Science Without soft Constraints.

Student	SMC
Completed first year students	0.9417
Completed second year students	0.9610
Completed third year students	0.9508
124 students have graduated in computer science	0.9417

The result shows much higher precision compared to experiment doing with online data. The reason is in offline data, students are graduated students and they followed and checked the list of courses in old curriculum with less number of subjects, while 58 students followed the new one. Unfortunately, in the new curriculum, there are new courses that these 58 students learned. However, as the above experiment was conducted, with more data to train and to test, adding soft constraints the course recommendation could bring better result.

Besides, the second choice to recommend courses for students in case they do not satisfy with the first one was also suggested in this model. Fig. 8 shows the cover of these two set of course recommendation on the real chosen course set.

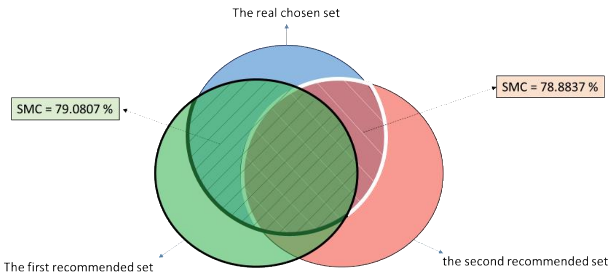

Figure 8. The more cover of 2 sets of recommended courses on the real chosen course set.

The second recommendation was made here with purpose that students could have various choice which is fit student’s constraints.

V. Implementation

Backend

We chose a pretty new and potential web technology Node.js to implement our web server. Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model which makes it lightweight, incredibly fast and efficient. Event-driven means that the server just performs when an event arises. Most I/O operation use non-blocking principle which means thread control is returned to caller with no waiting time for I/O operation. Those things make Node.js system has a high performance and scalability, suited for real-time application. Node is designed to build easily scalable network applications in the fastest way. Therefore, Node.js is one of the most reasonable technologies for our requirements.

After considering our web entities, MongoDB was taken into account in our system. MongoDB is a particularly flexible document-oriented database, which is easy to scale both vertically and horizontally. Hence, it’s a completely appropriate technology to build our web-database, most of data about students, lecturers, time table, education program, score, etc. These data would be updated frequently not just by adding new records but changing the schema structure. MongoDB is great for modeling many of the entities. It’s worldwide use by numerous number of big companies, some of them are Google, Facebook, Ebay, Expedia, etc. We created RESTful APIs using Expressjs (the most popular Node.js web framework) and mongoose to interact with MongoDB server.

Frontend

PRE system is built mainly for student-use so it’s must be an easy-to-use app, with friendly user interface, simple but still attractive. Flat Design has been one of the most influential trends in web design. Essentially, everything from user interfaces to other elements focuses on typography and colors rather than visual effects like 3D elements and shadows. Therefore, we use flat design as a major genre for our design. To be more effective in development and save the network resource we use Single Page Application model which seems to bring a lot of advantages. Single page apps are becoming increasingly popular in recent years. SPA is really fast because most of resource is only loaded once throughout the lifespan of application. Only the dynamic data is transmitted back and forth. The development is simplified and smoothed. There is no need to implement code to generate views on the server and numerous other advantages.

To implement system with propose requirements we need a SPA framework. There is an excessive number of modern framework but we found that Angular is the satisfactory one which is effectively, safety and helps us to create applications easily. With a large community where members act as testers make it be a more reasonable choice. This framework and supported libraries would be enough for building entirely a website. It is collaboration of JS with HTML and CSS.

Services

All AI services and tasks which have high computational complexity took place here. An isolated component, we use Java API for RESTful Web Services shorted for JAX-RS as a supporter to create web services according to REST architecture. It is where we define our core function for Study-path Recommendation as well as Student performance prediction. User would call those implemented API through the Node.js server. Thus, it is easy for control user interactions and leverage security of server-side.

VI. Conclusions and Future Work

In this paper, we have presented a course recommender system called PRE, a “Personalized Recommendation for Education” system which was mainly constructed on top of recommendation and some other machine learning techniques. The system’s AI modules used real data from educational affair at the University of Engineering and Technology for analysis and supervision. It analyses student behavior in the past and some extra information concerning course choices, recommending satisfactory research topics and predicting student performance. The accurate model of those functions helps managers providing better educational services. It frees students from tough decision as which course is the best for their interest and future career. These predicted and recommended results also provide them feedback soon, thus, we can prevent the students drop-out rate (or even expelling) every year.

The experiments yield quite good results; we achieved the best RMSE score of 1.669, the output of the student performance prediction model which uses the proposed hybrid approach. For study-path recommendation task, we achieved 0.7908 as the SMC score that run over total number of test information technology students. Nonetheless, those tasks still remain some limitations. One of them is that study-path recommendation task does not have a strong and meaningful measurement scale and it needs some students' feedbacks when using the system for improvement. In future, we would upgrade it based on real response from students. Furthermore, we would find out some unknown factors for enhancing PSP model strength. After the first release, based on student feedback we intend to upgrade to more functional and convenient website with satisfactory user experience. Moreover, we are in progress of developing the AI module which recommends student suitable research topics as well as supervisors. The work has archived a quite good result of the first step.

Acknowledgment

This work was fully supported by the Project QG.15.29 and Project QG.16.34 from Vietnam National University, Hanoi (VNU). Thi-Oanh Tran is the corresponding author (+84 362220684).

References

[1] A. Parameswaran, P. Venetis, and H. Garcia-Molina, "Recommendation systems with complex constraints: A course recommendation perspective," *ACM Transactions on Information Systems (TOIS)*, vol. 29, no. 4, 2011.

[2] A. Toscher and M. Jahrer, "Collaborative filtering applied to educational data mining," *KDD cup*, 2010.

[3] N. Thai-Nghe, T. Horv, and L. Schmidt-Thieme, "Personalized forecasting student performance," in *Proc. 11th IEEE International Conference on Advanced Learning Technologies (ICALT)*, 2011, pp. 412-414.

[4] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *ACM Sigmod Record*, vol. 22, no. 2, pp. 207-216, 1993.

[5] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, vol. 1215, pp. 487-499, 1994.

[6] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th International Conference on World Wide Web*, 2001, pp. 285-295.

[7] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*, pp. 291-324. Springer, 2007.

[8] M. Balabanović and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66-72, 1997.

[9] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, pp. 325-341, 2007.

[10] N. Bendakir and E. Aïmeur, "Using association rules for course recommendation," in *Proc. AAAI Workshop on Educational Data Mining*, vol. 3, 2006.

[11] Y. Lee and J. Cho, "An intelligent course recommendation system," *SmartCR*, vol. 1, no. 1, pp. 69-84, 2011.

[12] A. M. Shahiri and W. Husain, "A review on predicting student's performance using data mining techniques," *Procedia Computer Science*, vol. 72, pp. 414-422, 2015.

[13] P. L. Hsu, R. Lai, and C. C. Chiu, "The hybrid of association rule algorithms and genetic algorithms for tree induction: An example of predicting the student course performance," *Expert Systems with Applications,* vol. 25, no. 1, pp. 51-62, 2003.

[14] M. H. I. Shovon and M. Haque, "Prediction of student academic performance by an application of k-means clustering algorithm," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7, no. 1, pp. 292-295, 2014.

[15] P. Strecht, J. Mendes-Moreira, and C. Soares, "Merging decision trees: A case study in predicting student performance," in *Proc. Advanced Data Mining and Applications*, 2014, pp. 535-548.

[16] B. Şen, E. Uçar, and D. Delen, "Predicting and analyzing secondary education placement-test scores: A data mining approach," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9468-9476, 2012.

[17] P. Strecht, L. Cruz, C. Soares, J. Mendes-Moreira, and R. Abreu, "A comparative study of classification and regression algorithms for modelling students' academic performance," presented at International Educational Data Mining Society, June 2015.

[18] D. Kabakchieva, "Predicting student performance by using data mining methods for classification," *Cybernetics and Information Technologies*, vol. 13, no. 1, pp. 61-72, 2013.

[19] U. bin Mat, N. Buniyamin, P. M. Arsad, and R. Kassim, "An overview of using academic analytics to predict and improve students' achievement: A proposed proactive intelligent intervention," in *Proc. IEEE 5th Conference on Engineering Education (ICEED)*, pp. 126-130, 2013.

[20] P. M. Arsad and N. Buniyamin, "A neural network students' performance prediction model (NNSPPM)," in *Proc. IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 2013, pp. 1-5.

[21] J. F. Chen, H. N. Hsieh, and Q. H. Do, "Predicting student academic performance: A comparison of two meta-heuristic algorithms inspired by cuckoo birds for training neural networks," *Algorithms*, vol. 7, no. 4, pp. 538-553, 2014.

[22] T. Wang and A. Mitrovic, "Using neural networks to predict student's performance," in *Proc. Computer in Educations*, Jan. 2002, pp. 969-973.

[23] T. O. Tran, H. T. Dang, V. T. Dinh, and X. H. Phan. "Performance prediction for students: A multi-strategy approach," *Cybernetics and Information Technologies*, vol. 17, no. 2, pp. 164-182, 2017.

[24] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124-1137, 2004.

[25] T. H. Szymanski, "Max-flow min-cost routing in a future-Internet with improved QoS guarantees," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1485-1497, 2013.

Thi-Hai-Yen Vuong was born in Vietnam in 1994. She got her Bachelor of Information Technology from University of Engineering and Technology – Vietnam National University, Hanoi, Honors Program in 2016. Then, she was a master student at School of Information Science, Japan Advantage of Information and Science from 2016 to 2018. Now, she is a teaching assistant and research assistant at the University of Information and Technology, Vietnam National University, Vietnam. Her research interests include machine learning, data mining, natural language processing, and spoken dialogue systems.

Thi-Thu Trinh was born in Vietnam in 1994. She holds a Bachelor of Computer Science from University of Engineering and Technology – VNU in June, 2017, Hanoi, Vietnam. Now She is a service application developer in Sakura Information System company, Tokyo, Japan. She specializes in service development and web design. Making a "clever" web application is her desire in the near future.

Hong-Duyen Ha was born in Vietnam in 1996. She holds a Bachelor's degree of Information Technology from University of Engineering and Technology – VNU Honors Program in June,2018, Hanoi, Vietnam. Now she is a java web application developer in Synergix Technologies company, Ha Noi Office, Viet Nam. She specializes in web back-end development. Making a customizable, web-based Enterprise Resource Planning System is her desire in the near future.

Thi-Oanh Tran was born in Vietnam. She got her Bachelor of Information Technology and Master of Information Systems at University of Engineering and Technology – Vietnam National University, Hanoi in 2006 and 2009, respectively. Then, she obtained her PhD in Computer Sciences at JAIST (Japan) in 2014. Now, she works as a lecturer at International school, VNU-Hanoi. She has published 15 scientific papers, among which one paper is ISI-indexed, and the other two papers are SCOPUS-indexed. Her research interests include Data Mining, Natural Language Processing, and Business Intelligence.

Hieu-Xuan Phan received the BS and MS degrees in Information technology from College of Technology, Vietnam National University, Hanoi, in 2001 and 2003, respectively, and the PhD degree in Information Science from Japan Advanced Institute of Science and Technology JAIST) in 2006. He was a postdoctoral fellow by Japan Society for Promotion of Science (JSPS) at Graduate School of Information Sciences, Tohoku University, from 2006 to 2008. He was also a fellow researcher at the Center for Health Informatics, University of New South Wales. He is currently an associate professor at University of Engineering and Technology, Vietnam National University, Hanoi. His major research includes natural language processing, machine learning, web/text mining, and business intelligence.