BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **17**, No 2
Sofia • 2017 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2017-0024

Performance Prediction for Students: A Multi-Strategy Approach

Thi-Oanh Tran[1], *Hai-Trieu Dang*[2], *Viet-Thuong Dinh*[2], *Thi-Minh-Ngoc Truong*[2], *Thi-Phuong-Thao Vuong*[3], *Xuan-Hieu Phan*[2]

[1]*International School, Vietnam National University Hanoi, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*
[2]*University of Engineering and Technology, Vietnam National University Hanoi,144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*
[3]*Center of Education Testing, Vietnam National University Hanoi,144 Xuan Thuy, Cau Giay, Hanoi, Vietnam*
E-mails: oanhtt@isvnu.vn trieudh_58@vnu.edu.vn thuongdv_58@vnu.edu.vn ngocttm@vnu.edu.vn thaovtp@vnu.edu.vn hieupx@vnu.edu.vn

Abstract: *This paper presents a study on Predicting Student Performance* (*PSP*) *in academic systems. In order to solve the task, we have proposed and investigated different strategies. Specifically, we consider this task as a regression problem and a rating prediction problem in recommender systems. To improve the performance of the former, we proposed the use of additional features based on course-related skills. Moreover, to effectively utilize the outputs of these two strategies, we also proposed a combination of the two methods to enhance the prediction performance. We evaluated the proposed methods on a dataset which was built using the mark data of students in information technology at Vietnam National University, Hanoi* (*VNU*). *The experimental results have demonstrated that unlike the PSP in e-Learning systems, the regression-based approach should give better performance than the recommender system-based approach. The integration of the proposed features also helps to enhance the performance of the regression-based systems. Overall, the proposed hybrid method achieved the best RMSE score of 1.668. These promising results are expected to provide students early feedbacks about their* (*predicted*) *performance on their future courses, and therefore saving times of students and their tutors in determining which courses are appropriate for students' ability.*

Keywords: *Predicting student performance, academic system, hybrid approach, regression, recommender system.*

1. Introduction

Predicting Student Performance (PSP) has become one of the most common tasks in Educational Data Mining (EDM) [18, 20, 21]. It has drawn the attention of not only the EDM community but also the machine learning and data mining people (e.g., it is the topic of KDD challenge 2010 and a workshop at KDD 2011). This is the task

of predicting the performance of students on a specific course or degree based on their socio-demographic factors [23] and their performance on past course/degree [2] as well as the information when they interact with the tutoring/e-Learning system [28]. PSP can be built for e-Learning systems or academic systems. Most studies have investigated the task in e-Learning systems thanks to the availability of rich data. Not much research was dedicated to PSP in academic systems.

Nowadays, more and more universities/colleges are using credit systems in higher education. 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, academic credit can be gained by successfully completing a study course. Hence, choosing the right course is a critical decision and it is important to get it right, as it can impact students' future success. Students enrolled in a course they are not happy with, typically study it with low motivation. Unfortunately, when choosing elective courses students are usually uncertain because they do not know which ones are most suitable for them. One of the reasons is that they do not have sufficient background needed for selecting appropriate courses. Thus, the current solution is to make selection, supported by the direct guidance from their tutors/teachers. However, this process is rather expensive and further complicated in situations where the tutors/teachers background knowledge or information about the ability of their students is incomplete. Therefore, if we can predict the performance of students on unlearned courses, the students may know, at least, some information about their (predicted) performance on those courses, and may determine which ones are appropriate for their background and ability. Also, based on the predicted results, we can provide them early feedbacks, thus, we can prevent the dropping rate (or even expelling) every year.

Among work for PSP in academic systems, most of which mainly focused on PSP at the degree level, i.e., forecasting the student CGPA (Cumulative GPA) given a specific field of study (as an item) for each semester or academic year, etc., [7, 26, 28] or predicting the students' mark at the end of a university degree [1]. At the course level, Huang and Fang [11] predicts course performance using students' performance in prerequisite courses and midterm examination results. Unfortunately, at the time students choose courses, we do not have the information of midterm marks. Moreover, so far there is no systematic research on factors influencing the performance of students in a particular course, especially in academic systems where we do not have much information available. Previous work on PSP in e-Learning systems mostly suggested that some academic performance is needed for good results and that socio-demographic factors might be less relevant [1, 9]. We, thus, need to use additional useful information about students' academic performance to effectively predict their performance. In this work, we propose using the available information of not only prerequisite courses (as in the work of [11]) but also all completed courses, total cumulative GPA, GPA of previous semesters, etc., to predict course performance of students. In more details, we propose a method of setting relations between courses, which are based on courses' attributes (see Section 3.4 for more

details). This information will be used as features to build our regression-based predictors.

Another direction for PSP in academic systems is the strategy of considering the task as a rating prediction task in recommender systems, as previously proposed for the task in e-Learning systems [27, 29]. This strategy predicts the mark of a student on a particular unlearned course based on the performance of other students, who share the same past performance patterns with the student whom the prediction is for. This strategy is also carefully investigated in this work.

To effectively use the results of these two strategies, we also propose a simple hybrid method to combine the outputs of previous systems in order to enhance the performance of the final prediction system. The experimental results are reported based on a dataset which is built from the data of 1268 undergraduate students in the field of Information Technology (IT) at Vietnam National University, Hanoi. The main contributions of this work are as follows:

- Building a dataset consisting of students, completed courses, and their scores in an academic system.
- Investigating and formulating the task of PSP in academic systems using two strategies which are based on recommender system and traditional regression techniques.
- Designing course-related skills in academic systems, which will be used as features in regressions-based models to improve their performance.
- Proposing a hybrid method to effectively combine the best outputs of these two strategies in order to enhance the performance of the final system.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the methods used to address the task including how to formulate the PSP task as regression and rating problems, as well as a simple combination method. Section 4 describes the dataset, the experiment settings and the experimental results. Section 5 discusses and analyzes some typical errors caused by the final system. Finally, contributions and conclusions are given in Section 6.

2. Related work

Prediction models proposed for PSP can be categorized into two main strategies.

In the first strategy, authors usually formulate it as a classification or regression problem and use some typical machine learning algorithms such as SVM [12, 17, 25], linear regression [22], decision tree [7, 15, 24], ANN [3], etc., to build and test models at both course and degree levels. For example, Asif, Merceron and Pathan [1] tried to predict performance of students at the end of a university degree at an early stage of the program by using pre-university marks and marks of 1st and 2nd year courses with a reasonable accuracy. Golding and McNamarah [8] determined the relationship between students' demographic attributes, qualification on entry, aptitude test scores, and performance in the 1st year and their overall performance in the program. Zimmermann et al. [30] examined the statistical relationship between B.Sc. and M.Sc. achievements. Thai-Nghe Janecek and Haddawy [26] predicted students' performance in two different case studies of

Can Tho University (CTU) and the Asian Institute of Technology (AIT). In the first case, they predicted GPA at the end of the 3rd year by using the students' records including English skill, field of study, faculty, gender, age, family, job, religion, etc., and the 2nd year GPA. In the second case, they used students' admission information (including academic institute, entrance GPA, English skill, marital status, Gross National Income, age, gender, TOEFL score, etc.) to predict the GPA of the master students at their first year. Another work predicted students' graduate level performance by using undergraduate achievements [30]. At the course level of academic systems, Huang and Fang [11] predicted course performance using students' performance in prerequisite courses and midterm examinations. Relating to features used, there are also various types including past academic performance of students, socio-demographic factors, records of students. However, there is no systematic research on factors influencing the students' performance in a particular course so far, especially in the academic system where we do not have much available information.

In the second strategy, the PSP task can be seen as a rating prediction problem in recommender systems [28, 29]. 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. Specifically, Toscher and Jahrer [29] adopted k-NN and matrix factorization for the KDD cup competition. The resulting solution ranked number three in the KDD Cup 2010. Thai-Nghe and Horvath [28] chose tensor factorization methods to model sequential/temporal effects in students' knowledge acquisition progress. To validate this strategy, the authors compare recommender system techniques with traditional regression methods such as logistic/linear regression by using educational data for intelligent tutoring systems. In this research, authors showed that the proposed approach gave better performance in comparison to the traditional regression/classification in performance prediction of e-Learning systems.

Most previous work focuses on PSP in 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 PSP in order to help them choosing the right course is becoming more and more important. 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. Another advantage is to provide them early feedbacks; thus, we can prevent the students dropping every year. With these important goals, we have to investigate additional features that might influence the performance of students in a particular course. Some features which were investigated in previous work will not be included (e.g., the information of mid-term examinations as proposed in [11] is not available at the time the students choose right courses). To learn and test the prediction models, we investigate two strategies that considered the PSP task as a regression problem and a rating prediction problem in recommender systems (which were successfully

done for the PSP task in e-Learning systems [28, 29]. About the features, we propose an additional feature set based on courses-related skills to effectively improve the performance of regression-based prediction models. In addition, to take advantages of the outputs of these two strategies, we will also propose a simple yet effective hybrid method using linear combination to enhance the performance of the final prediction system.

3. PSP as regression and collaborative prediction problems

Let X be a set of students, C be a set of subjects/courses that students should take, and S be a range of possible marks/scores ($S \in [1, \dots, 10]$). In the supervised setting, the PSP task is formally described as follows.

Given the set of training data, we need to find:

(1) $$\hat{s}: X \times C \to R,$$

such that the Root Mean Square Error (RMSE) measure of an estimator $\hat{s}$ with respect to an estimated parameter s is minimum on the test data. In the next sections, we will present how to recast the task as a regression/classification problem and a rating prediction problem in recommender systems.

3.1. PSP as regression and classification problems

This section shows how to map PSP to a regression/classification problem and then describes some typical algorithms such as Linear Regression (LN) [10], Artificial Neural Networks (ANN) [13], Decision Tree (DT) [19], and Support Vector Machines (SVMs) [4]. These are also main methods used in this work. In this strategy, a set of mathematical formula was used to describe the quantitative relationships between the outputs and the inputs. The prediction is accurate if the error between the predicted and actual values is within a small range.

In principle, this can be considered as a regression problem. Similarly, if the predicted values are categorized (e.g., $S \in \{A, B, C, D, E\}$), the task would be considered as a classification problem. In the following sub-sections, we will briefly describe some efficient machine learning methods which are used in this paper.

3.1.1. Linear regression

Linear Regression (LR) is a simple yet effective predictive analysis. It is used to describe and explain the relationship between one dependent variable y and one or more independent variables $x_i\{i = 1, \dots, n\}$. In our setting, the dependent variable is the score that students earned/will earn in a specific course. The independent variables are features describing the characteristics of students and the courses that students completed.

Given a dataset $\{y_i, x_{i1}, \dots, x_{ip}\}_{i=1}^{n}$ of n samples, a model of LR assumes that the relationship between y_i, and the p-vector of regressors x_i is linear. This relationship is modeled through a disturbance term or error variable ε_i – an unobserved random variable that adds noise to the linear relationship between the dependent variables and regressors. Thus the model takes the following form:

(2) $$y_i = \alpha_1 x_{i1} + \cdots + \alpha_p x_{ip} + \varepsilon_i, i = 1, \ldots, n.$$

The parameters of the model $\alpha_1, \ldots, \alpha_p$ will be estimated on the training dataset.

3.1.2. Artificial neural networks

Artificial Neural Networks (ANNs) are a computational approach which is based on a large collection of neural units loosely modeling how the brain solves problems. ANNs are structured in layers. Layers are made up of a number of interconnected "nodes" which imitate biological neurons of human brain. The nodes can take the input data via the "input layer", which communicates to one or more "hidden layers" where the actual processing is done. The hidden layers then link to an "output layer" where the answer is output.

Fig. 1 illustrates a typical ANN with one input layer, one hidden layer and one output layer. The output at each node is called its activation or node value. Each link is associated with its weight. ANNs are capable of learning, which takes place by altering weight values.

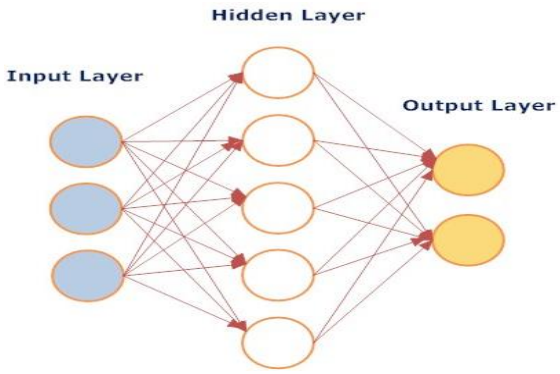


Fig. 1. An example of a simple ANN structure

3.1.3. Decision tree

Decision Trees (DTs) are classic algorithms, which are organized in a tree-like structure in which each internal node represents a 'test' on an attribute. For example, one node can test what is the required math ability to study a particular course. Each branch represents the outcome of the test and each leaf node represents a class label (e.g., predicting score taken after computing all attributes). The paths from root to leaf represent classification rules. The goal is to achieve perfect classification with minimal number of decision, although not always possible due to noise or inconsistencies in data.

The core algorithm for building decision trees called ID3 [19] which employs a top-down, greedy search through the space of possible branches with no backtracking. The main challenge while building the tree is to decide on which attribute to split the data at a certain step in order to have the 'best' split. To do this, we use the concept of Information Gain (IG), which measures the difference between the entropy before and after a decision. In regression setting, the ID3 algorithm uses standard deviation reduction as a replacement of IG to construct a decision tree.

3.1.4. Support vector machines

The Support Vector Machines (SVM) were successfully applied not only to classification problems but also to the case of regression in many areas. The algorithm can be stated as follows:

Suppose we are given the training data $\{(x_i, y_i), \ldots, (x_n, y_n)\} \in X \times R$ where X denotes the space of the input patterns - for instance, difficulty levels (ranging from 1 up to 5) of a specific course. In ε-SV regression Vapnik, the goal is to find a function $f(x)$ that has at most deviation from the actually obtained targets y_i for all the training data, and at the same time, is as flat as possible. SVMs rely on defining the loss function that ignores errors, which are situated within the certain distance of the true value. Fig. 2 shows an example of one-dimensional linear regression function and non-linear regression function with epsilon intensive band.

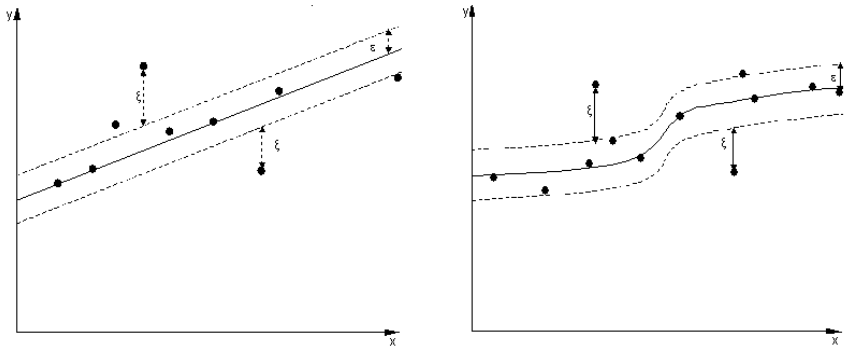


Fig. 2. One-dimensional linear regression (on the left-hand side) and non-linear regression functions (on the right hand side) with epsilon intensive band

In the case of linear functions, f taking the following form:

$$f(x) = \langle w, x \rangle + b \text{ with } x \in X, \quad b \in R, \tag{3}$$

where $\langle ., . \rangle$ denotes the product in X. To ensure *Flatness* in Equation (3), we can minimize the Euclidean norm, $\frac{1}{2}\|w\|$, which subject to the two following constraints:

$$\begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon, \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon. \end{cases} \tag{4}$$

Moreover, we can use the dual formulation to provide the key for extending SV machine to non-linear functions. In reality, we can use a standard dualization method utilizing Lagrange multipliers as described in (Fletcher, 1989)

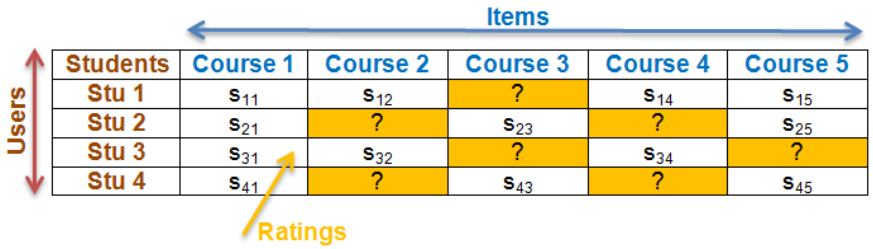
3.2. PSP as a rating prediction in a recommender system

This section shows how to map PSP to a rating prediction task in collaborative filtering and then briefly describes the CF technique applied in this scenario.

Recently, recommender systems [16] have become much more popular, and are being applied in many areas such as video-on-demand, music, news, research article, e-commerce, etc. They have also been utilized in Technology Enhanced Learning [5] whose aim is to design, develop, and evaluate socio-technical innovations for various

kinds of learning and education. Some typical examples include the work of Manouselis et al. [14] that focused on recommending learning contents to the learners in e-Learning systems, the work of Garcia et al. [6] focusing on recommending course enrollment, etc.

Since the competition in the Knowledge Discovery and Data Mining Cup 2010, a new application of recommender systems in student modeling and PSP tasks has been introduced. One of the winners [29] pointed out that there is a mapping between PSP and the rating prediction task in Collaborative Filtering (CF) where students, courses, and marks would become users, items, and rating values, respectively. Authors chose the method of CF, such as k-NN and matrix factorization [29], tensor factorization models [28] to build prediction models. Fig. 3 shows the similarity between the PSP task and the rating prediction task in recommender systems.



Students	Course 1	Course 2	Course 3	Course 4	Course 5
Stu 1	s_{11}	s_{12}	?	s_{14}	s_{15}
Stu 2	s_{21}	?	s_{23}	?	s_{25}
Stu 3	s_{31}	s_{32}	?	s_{34}	?
Stu 4	s_{41}	?	s_{43}	?	s_{45}

Fig. 3. Similarity between a PSP task and a rating prediction task in recommender systems(s_{ij}: the score of student i taking course j)

The underlying idea behind the CF technique is to calculate students' scores of unlearned courses based on the scores of students, who share the same past performance patterns with students whom the prediction is for.

Consider student x to whom we want to predict his/her score on a specific unlearned course. We need to find a set of other students (called set N) whose performances on completed courses are similar to the performance on these completed courses. These students are called the neighborhood of student x. The key trick is to calculate the similarity between students. To do this, there are several options, such as Jaccard similarity, cosine similarity, centered cosine similarity (also known as Pearson Correlation), etc. For examples, if we use Pearson correlation to calculate the similarity $\mathrm{sim}(x, y)$ between two students x and y then the formula is as follows:

(5) $$\mathrm{sim}(x,y) = \frac{\sum_{i\in C}(s_{x,i}-\overline{s_x})(s_{y,i}-\overline{s_y})}{\sqrt{\sum_{i\in C}(s_{x,i}-\overline{s_x})^2}\sqrt{\sum_{i\in C}(s_{y,i}-\overline{s_y})^2}},$$

where $s_{x,i}$ is the score of student x for a completed course i, C is the set of courses studied by both students x and y, and $\overline{s_x}$ is student x 's average scores.

To predict the performance of student x on an unlearned course i, $\hat{s}_{x,i}$, we can weight the average scores by the similarity values as shown in Formula 6. In our setting, possible similarity values between -1 and 1, and scores value from 0 to 10.

(6) $$\hat{s}_{x,i} = \frac{\sum_{y\in N}\mathrm{sim}(x,y)s_{y,i}}{\sum_{y\in N}\mathrm{sim}(x,y)}.$$

3.3. The hybrid method

In this section, we present a proposed hybrid method. In this method, we combined the outputs from the collaborative filtering-based system and the regression-based system using a linear combination method as shown in Equation (7). Following this formula, the predicted score of student i taking course j is calculated as follows:

$$\text{ScoreHybrid}_i^j = \alpha \times \text{ScoreCF}_i^j + \beta \times \text{ScoreRe}_i^j, \quad \text{s.t, } \alpha + \beta = 1 \tag{7}$$

where ScoreCF_i^j: the predicted score of student i taking course j using the CF-based method; ScoreRe_i^j: the predicted score of student i taking course j using the regression-based method. In experiments, we choose the best regression model – the model uses SVMs with the Tr-All training method and integrating all proposed features – to make combination. The parameters of α, β will be estimated using a development set.

3.4. The features

This section intensively discusses important factors that might affect the performance of the PSP task in the regression/classification settings.

There are various attributes types used for PSP in tutoring systems including past academic performance of students [1, 11], socio-demographic factors [15], and records of students [11]. Most works showed that previous marks/scores can be used to predict the scores in a course with high accuracy [1, 11]; and that socio-demographic factors might be less relevant [1, 9]. Moreover, some socio-demographic factors (e.g., family supports, extra-curricular activities, social interaction network, etc.), of students in Vietnamese academic systems are difficult (or impossible) to collect. In this work, therefore, we focus on factors of past academic performance and records of students to predict students' scores on unlearned courses. We collected the available information of students including gender, total cumulative GPA, GPA of previous semesters, average scores of prerequisite courses, semesters that courses were taken.

Table 1. Detailed set of skills required for each course

No	Attributes	Values	Notes
1	Difficult levels	1, 2, 3, 4, 5	The higher, the more difficult
2	Types of courses	Seven major groups of training program 2012	
3	Ability of learning by heart	1, 2, 3, 4, 5	The higher, the better
4	Math knowledge	1, 2, 3, 4, 5	
5	English knowledge	1, 2, 3, 4, 5	
6	Testing methods	Writing, interviewing, practicing	
7	Major fields	One of four major fields in IT	Computer Science, Information Systems, Computer networks, and System technology
8	Programming abilities	1, 2, 3, 4, 5	The higher, the better
9	Group working abilities	Yes/No	
10	Rates of theory hours	*x*/3	$x \in [0, \ldots, 3]$
11	Rates of practice hours	*x*/3	$x \in [0, \ldots, 3]$
12	Avg. scores of pre-requisite courses	$[0, \ldots, 10]$	

Beyond the limitation of previous work, we also investigate another type of attributes that might affect the prediction. It is assumed that there are some required skills to do a task. Specifically, each course requires some skills (e.g., English ability, programming ability, mathematic background, teamwork skills, communication skills, etc.), to perform it. These requirements are actually hidden in students' performances on completed courses (the higher the performance of a course, the better the skills related to that course, e.g., if scores of English courses of a student are high, English skills of that student are also good). If students' skills are good, the performances of courses required those skills are likely to be high. Therefore, it is reasonable to use the information of past courses' performance to predict the performance on unlearned courses. The problem is that we have to build a reasonable set of skills required for courses. To do this, we ask the helps of human experts in specific fields (including people who design the courses, some lecturers and students studying these courses) to design a required skill set for courses in a particular Training Program (TP).

To implement, we had two experts and two graduated students to compose the skill list and then mark values for each course in the TP of the IT field at VNU-UET. Table 1 shows the detailed attributes including 12 main ones: *difficulty levels, types of courses, ability to learn by heart, math knowledge, English knowledge, testing methods, major fields, programming abilities, group working abilities, rates of theory hours/practice hours, and average scores of pre-requisite courses*.

4. Experiments

4.1. Dataset collection

With the support of the Student and Academic Affairs of a national university in Vietnam, we collected the data including the information of 1268 undergraduate students following the standard IT program in seven years (from K52 to K58). In these seven years, there are three standard TP published in 2007, 2009 and 2012, respectively. These TPs mostly match each other, but they still have some small modifications. To keep up-to-date, we chose the latest TP released in 2012. This program includes 78 subjects categorized into six groups (*including* (1) *General Education Knowledge*, (2) *Basic Professional Knowledge*, (3) *Basic Professional Knowledge of IT and ET*, (4) *Professional Knowledge–Compulsory*, (5) *Professional Knowledge–Complementary, and* (6) *Targeted Elective Courses*). Therefore, we had to standardize the dataset of the two previous TP based on this program. For students following the two previous programs, if their completed courses are not exactly coincident with the ones in the latest one, we performed modifying them as follows:

- Soft skill courses: skip them because they did not contribute to the final student performance.
- Changes in course codes: use the codes in the latest TP.
- Changes in course names: map into the most similar one in the latest TP.
- Changes in the number of course credits: choose the new credit numbers of the latest TP.

- Combining of separated courses: get the average scores over separated subjects.
- Splitting courses: get the scores of those subjects to fill in scores of each split subject.
- Adding new courses, removing old ones based on the latest TP.

Finally, we obtained the dataset including 1268 students along with the information of their personal information, scores on completed courses, and the course information. The details of students' information include student names, student IDs, genders, date-of-births, and scores achieved at completed courses, learning times of each course, and semesters/years which the courses were taken. The information of courses includes course names, course codes, credit numbers, and prerequisite courses.

4.2. Experimental setups

For each course in the latest TP, we built a separate predictor for it. We randomly split the dataset of each course into two disjoint set. The first set consists of about 10% of that dataset, called development set, for choosing parameters of the hybrid method. The remaining 90% is used for building and testing the predicting model.

To train and test the model, we performed 10-fold cross validation test. In this setting, all students taking that course will be randomly partitioned into 10 equal folds. At each round, a fold will be used to test and the 9 remaining folds will be used to train the model. The performance measures are then averaged over 10 loops.

In building predicting models, we performed two methods of getting the training data. Assume that we are building the predicting model for a given course c_i, for each student x_j studied c_i we create training instances as follows:

- Tr_All: getting data of all completed courses that x_j has already taken. These courses can be taken before, after, and at the same time that c_i was taken.
- Tr_Sub: getting data about only completed courses which were accomplished before the time c_i was taken.

For testing data, we only got data about courses accomplished before the testing course c_i was taken. This is due to the fact that at the time of predicting the score for c_i, student x_j only possesses the score data about the completed courses which were already finished.

We built a separate predictor for each course, measured RMSE scores, and then got the average results on all 73 courses of latest TP. In order to avoid cold-start problems of CF models, we left the very first beginning 5 courses of the first term for building training instances. This is due to the fact that in the first term, students usually have no choice of choosing courses that they want. To learn and test prediction models, we used different tools for different proposed methods. For the CF approach and some baselines, we used the MymediaLite tool which was developed at University of Hildesheim. For other regression methods, we used the Weka tool to run machine learning methods of neural networks, decision tree, and linear regression. The remaining SVM method was run by using the Libsvm tool.

The performance of prediction systems is measured by RMSE score. This is a frequently used measure of the differences between student scores predicted by a model and the real scores actually obtained. The RMSE of a score (mark) estimator $\hat{s}$ with respect to an estimated score s is defined as the square root of the mean square error as shown in the following formula:

$$\text{RMSE}(\hat{s}) = \sqrt{E((\hat{s} - s)^2)} = \frac{1}{n}\sum_{i=1}^{n}(\hat{s}_i - s_i)^2, \quad (8)$$

where n is the number of students need predicting scores for a given course c_i.

4.3. Experimental results

To evaluate the performance of proposed models, we got the averaged RMSE scores over courses. We measured on two types of courses:

- All courses: consisting of all courses in the training program, both compulsory courses and elective courses.
- Elective courses: consisting of only elective courses. This information will be more meaningful for students in choosing elective courses to study.

4.3.1. Experimental results using the CF strategies and some baselines

In this section, we present experimental results using the CF approaches compared with some baselines as proposed in [28]. Three baseline methods are used including student average, course average, and global average. Table 2 showed that the CF approach using matrix factorization techniques outperforms two baselines on both methods of getting training data. However, it is competitive with the baseline of student average. For all courses, the CF approach got the best results. However, for elective courses, the baseline of student average got the higher performance. Overall, the CF approach still yields the lowest RMSE of 1.915 for all courses, and 2.022 for elective courses when using the Tr_All training method. It can be said that for this task in academic systems, the CF approach is not as effective as it is for this task in e-Learning systems. Experimental results also indicated that using all completed courses of students to train the model yields better performance than using only courses studied before a given predicting course. In other words, it has already enriched the predicting model by providing more information.

Table 2. RMSE measures on two ways of getting training data using the CF and some baselines

Approach	Methods	All Courses		Elective Courses	
		Tr_All	Tr_Sub	Tr_All	Tr_Sub
Baselines	Student Average	1.923	1.929	**2.020**	**2.025**
	Course Average	1.958	2.098	2.045	2.200
	Global Average	2.082	2.098	2.183	2.200
CF	Matrix Factorization	**1.915**	**1.925**	**2.022**	2.028

Experimental results also expressed that the performance of predictors on elective courses is worse than on all courses. This may be due to the fact that the number of students who takes compulsory courses is larger. Especially, some elective courses have a very small amount of student studying them (e.g., for the course

INT3207, there are only 185 students taking that course, while the compulsory course POL101, most students (1134 students among 1268 students) study it).

4.3.2. Effects of getting additional skills-related features using regression models

To estimate the effect of additional skills-related features, we conducted two kinds of experiments. In the first experiment, we did not use the additional feature set. The only available information used to predict student performances includes of students' gender, course ID, scores, semesters taken, CGPA, GPA of previous semesters, and average scores of pre-requisite courses. In the second experiment, we add the additional feature set as proposed in Section 3.4. We performed experiments on two methods of getting training data using four strong machine learning methods (as described in Section 3.1). The experimental results illustrated in Table 3 and Table 4 showed that using skills-related feature was indeed effective.

Table 3. All Courses: RMSE measures on two methods of getting training data using different machine learning methods for regression problems

Skills-related Features	Tr_All				Tr_Sub			
	LR	ANN	DT	SVM	LR	ANN	DT	SVM
No	1.979	2.030	1.994	1.907	2.021	2.103	2.022	1.977
Yes	1.883	1.939	1.845	**1.705**	2.143	2.116	2.056	**1.727**

Table 4. Elective Courses: RMSE measures on two methods of getting training data using different machine learning methods for regression problems

Skills-related Features	Tr_All				Tr_Sub			
	LR	ANN	DT	SVM	LR	ANN	DT	SVM
No	2.121	2.160	2.115	2.054	2.154	2.217	2.126	2.114
Yes	1.994	2.046	1.848	**1.791**	2.320	2.120	2.092	**1.825**

The experimental results also strengthen the conclusion that using all completed courses of students in the training set yields better performance than using only a subset of them over all four algorithms.

4.3.3. Estimating the effect of each feature in the proposed feature set on regression models

In this sub-section, we performed feature selection to estimate the effect of each feature on regression models as well as selecting a subset of relevant features for use in model construction. We used the traditional statistics method, the most popular form of feature selection is stepwise regression, to do feature selection. It is a greedy algorithm that adds the best feature (or deletes the worst feature) at each round. We chose the best model of SVMs performing on elective courses to estimate the effectiveness of each proposed feature.

Fig. 4 visualizes the experimental results. As you can see that, the more proposed features added to the model, the better the performance of the regression model. This conclusion is true until the last feature, *learning-by-heart ability*, is added. The reason might be this attribute is not effective in this case and also quite difficult to exactly quantify for each course in the training program.

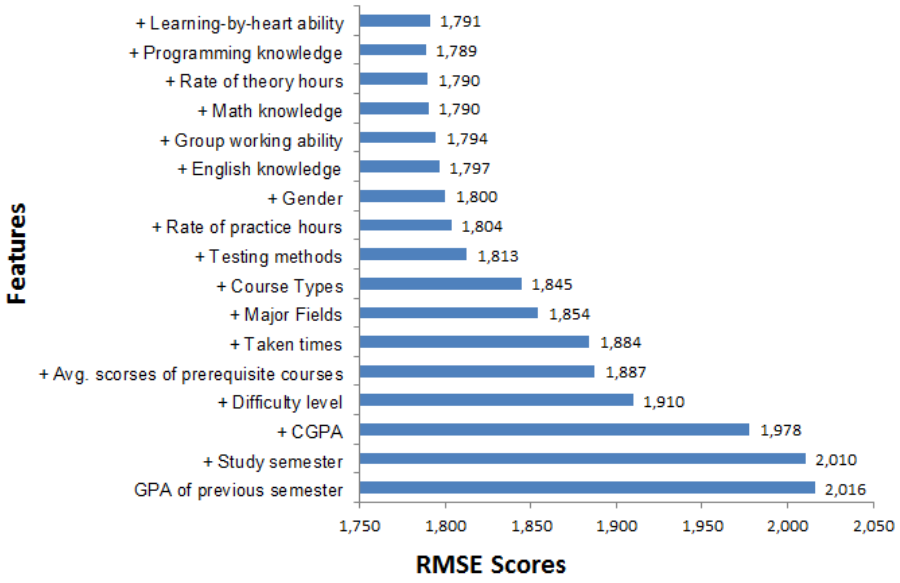


Fig. 4. Experimental results of incrementally adding each feature into the regression model

4.3.4. Combining CF and regression strategies

We chose the best performances of each strategy to perform combination. In the regression strategy, the best output was of the method which is built based on SVM algorithms using the Tr_All training method and adding all skills-related features (except for the *learning-by-heart* ability due to its inefficiency). In the CF strategy, the output of the method using matrix factorization was chosen. Then, we conducted combining these prediction outputs to enhance the performance of the final system (see Section 3.3).

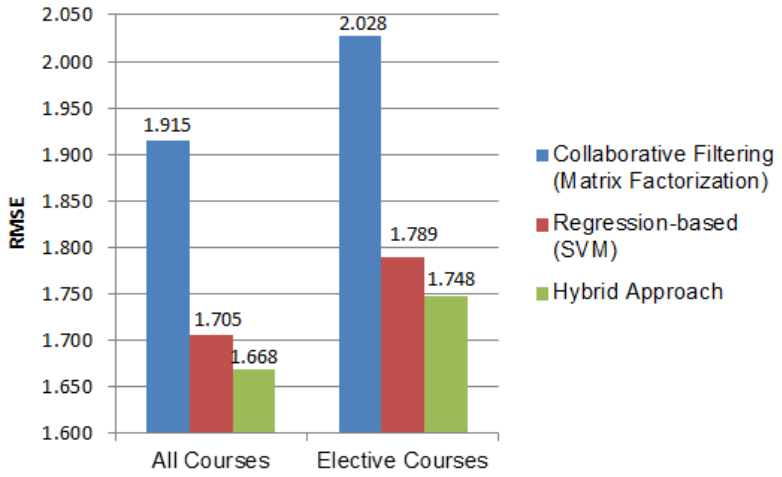


Fig. 5. Hybrid method results using the best outputs of CF and regression strategies

In experiments, we varied the parameter α between 0 and 1 (with steps of 0.1) and measured RMSE scores. The parameters of the best RMSE score on a development set are used to combine the outputs on testing data. The experimental results in Fig. 5 showed that the best combination of α(0.3) and β(0.7) yields the best RMSE score of 1.668. On elective courses, we also got the lower RMSE score of 1.748 in comparison with each individual approach. These results proved that this simple hybrid method is quite effective for the PSP task in academic systems.

4.3.5. Comparing RMSE measures among different knowledge groups

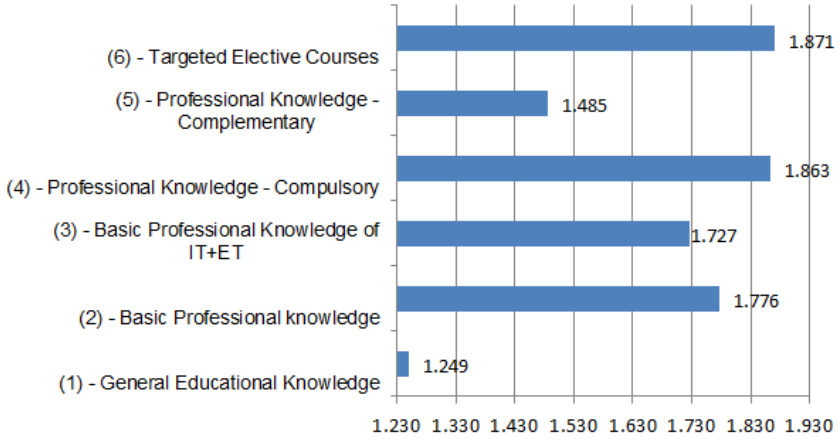


Fig. 6. Experimental results of the PSP task measured on each knowledge group

We performed some statistics on different knowledge groups in the TP 2012 based on the output of the best final prediction system – the hybrid approach. Experimental results on different knowledge groups were illustrated in Fig. 6. It is shown that the 1st group and the 5th group have the best performances. Courses (e.g., English 1-2-3, Algebra, Mathematical Analysis, Marxist-Leninist theory 1-2, Optimization, etc.), in these groups usually require high abilities of English, learning by heart, and math knowledge. It can also be said that there is no much difference between these groups.

5. Error analysis

This section discusses some typical errors caused by the final predicting system. Observing 12 courses having highest RMSE scores (see Table 5) we realized that they are mostly elective courses except the first one (courseID 10 – Algebra).

There might be some reasons for this: Firstly, the training data size of these elective courses is usually small; secondly, scores of these courses are quite polarized. An example is illustrated in Fig. 7 which shows the score distribution of INT3217 (the RMSE score is highest at 3.622) and INT3506 (the RMSE score is lowest at ~0.910). We can see that in INT3207, most student scores fall into the high range of 8-10, while in INT3506 the score ranges are quite uniform.

Table 5. Courses having high RMSE scores (greater than 2)

No	Course ID	Course Code	RMSE
1	10	MAT1093	2.051
2	29	INT3306	2.188
3	31	INT3507	2.610
4	43	INT3108	2.129
5	51	INT3217	3.622
6	52	INT3301	2.070
7	56	INT3307	2.289
8	58	INT3310	2.094
9	60	INT3505	2.497
10	62	INT3401	2.032
11	64	INT3404	2.158
12	70	INT3405	2.461

Moreover, we also observed prediction scores of students whose real scores are quite different from the predicted ones. There are many reasons for this high difference. For example, in semester 7, the student 10020458 studied the courseID 51 and got 7.2, but the system predicted 4.8. At that semester, this student studied this course along with seven other courses, among which there are four courses studied again to improve scores and the courseID 51 is one of them. Another reason might be the overload status he could encounter when studying too much courses in one semester (On average, students only study about 5 courses at the same time).

In reality, there is a fact that with the same course, score distributions among different lectures are quite different. This factor was not captured by our prediction model.

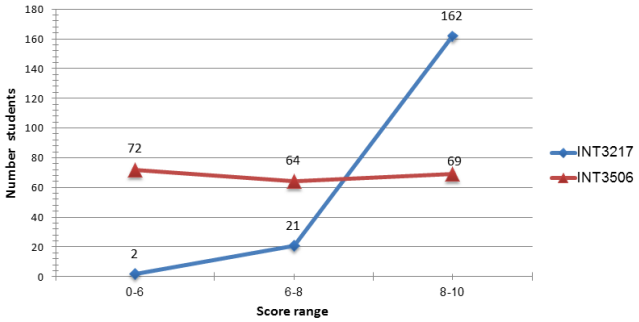


Fig. 7. Final score range distribution of two courses: INT 3217 and INT3506

5. Conclusions and future work

We have presented a study on PSP in academic systems. The accurate prediction of student performance not only helps managers providing better educational services, but also helps students foreseeing some information about their (predicted) performances on those courses, and may determine which ones are appropriate for their abilities/preferences. These predicted results also provide them early feedbacks, thus, we can prevent the students dropping (or even expelling) every year. The two most common strategies to this task were carefully investigated: the traditional one,

which recast the task as a regression/classification problem, and the recently proposed one for the PSP task in e-Learning systems, which maps the task as a rating prediction task in recommender systems. To effectively apply the first strategy, we proposed an additional feature set based on courses-related skills to improve the performance. Moreover, we also proposed a hybrid method based on linear combination to improve the performance of the final predicting system.

The experiments were carried out using a dataset which was built based on the score data of IT students at Vietnam National University, concerning 1268 students and 73 related courses (the dataset would be released once this work has been published). We found that for this PSP task, unlike in e-Learning systems, the later strategy based on recommender systems was not able to beat the traditional regression strategy for this task in academic systems. In the first approach, the algorithms of SVMs yield the best results. However, there is no significant difference in performance between the algorithms. The proposed additional feature set also clearly improved the performance of the regression-based approach. Overall, we got the best RMSE score of 1.668, the output of the system which uses the proposed hybrid approach.

In the future, as a complement of the problems studied in this work, it should be interesting to predict an interval for a score (e.g., intervals of {A, B, C, D, E}). We will also integrate these results into our personalized recommender system for education. Moreover, on the base of the performance prediction results, we are building course recommendation systems which recommend the most suitable courses for each student in respect to both the personal profile, preferences/ weaknesses, careers' targets of each student and the courses' requirements.

***Acknowledgements*:** This work was supported by the project QG.15.29 from Vietnam National University (VNU), Hanoi.

References

1. A s i f, R., A. M e r c e r o n, M. K. P a t h a n. Predicting Student Academic Performance at Degree Level: A Case Study. – International Journal of Intelligent Systems and Applications, Vol. **7**, 2015, No 1, pp. 49-61.
2. C h e n, S. M., T. K. L i. Evaluating Students' Learning Achievement by Automatically Generating the Importance Degrees of Attributes of Questions. – Expert Systems with Applications, Vol. **38**, 2011, No 8, pp. 10614-10623.
3. C h e n, J. F., H. N. H s i e h, Q. H. D o. Predicting Student Academic Performance: A Comparison of Two Meta-Heuristic Algorithms Inspired by Cuckoo Birds for Training Neural Networks. – Journal of Algorithms, Vol. **7**, 2014, No 4, pp. 538-553.
4. C o r t e s, C., V. V a p n i k. Support-Vector Networks. – Machine Learning, Vol. **20**, 1995, No 3, pp. 273-297.
5. D r a c h s l e r, H., K. V e r b e r t, O. C. S a n t o s, N. M a n o u s e l i s. Panorama of Recommender Systems to Support Learning. Recommender Systems Handbook. Part III. New York, Springer, 2015, pp. 421-451.
6. G a r c i a, E., C. R o m e r o, S. V e n t u r a, C. D. C a s t r o. An Architecture for Making Recommendations to Courseware Authors Using Association Rule Mining and Collaborative Filtering. – In: User Modeling and User-Adapted Interaction. Vol. **19**. No 1. Springer Netherlands Publisher, 2009, pp. 99-132.

7. G r a y, G., C. M c G u i n n e s s, P. O w e n d e. An Application of Classification Models to Predict Learner Progression in Tertiary Education. – In: Advance IEEE International Computing Conference (IACC'14), 2014, pp. 549-554.
8. G o l d i n g, P., S. M c N a m a r a h. Predicting Academic Performance in the School of Computing and Information Technology (SCIT). – In: Proc. of 35th ASEE/IEEE Frontiers in Education Conference, S2H, 2005.
9. G o l d i n g, P., O. D o n a l d s o n. Predicting Academic Performance. – In: Proc. of 36th Annual Conference in Frontiers in Education, 2006, pp. 21-26.
10. H i l a r y, L. S. Studies in the History of Probability and Statistics. XV Historical Development of the Gauss Linear Model. – Journal of Biometrika, Vol. **54**, 1967, No 1/2, pp. 1-24.
11. H u a n g, S., N. F a n g. Predicting Student Academic Performance in an Engineering Dynamics Course: A Comparison of Four Types of Predictive Mathematical Models. – Computers and Education, Vol. **61**, 2013, pp. 133-145.
12. K a b a k c h i e v a, D. Predicting Student Performance by Using Data Mining Methods for Classification. – Cybernetics and Information Technologies, Vol. **13**, 2013, No 1, pp. 61-72.
13. M a c k a y, D. J. C. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2012. 640 p.
14. M a n o u s e l i s, N., H. D r a c h s l e r, R. V u o r i k a r i, H. H u m m e l, R. K o p e r. Recommender Systems in Technology Enhanced Learning. 1st Recommender Systems Handbook. Publisher, Berlin, Springer, 2010, pp. 387-415.
15. M a t, U. B., N. B u n i y a m i n, P. M. A r s a d, R. K a s s i m. An Overview of Using Academic Analytics to Predict and Improve Students' Achievement: A Proposed Proactive Intelligent Intervention. – In: Proc. of International IEEE Conference on Engineering Education (ICEED'13), 2013, pp. 126-130.
16. M e l v i l l e, P., V. S i n d h w a n i. Recommender Systems. Encyclopaedia of Machine Learning Book. – New York, Springer, 2011, pp. 829-838.
17. O s m a n b e g o v i c, E., M. S u l j i c. Data Mining Approach for Predicting Student Performance. – Economic Review, Vol. **10**, 2012, No 1, pp. 3-12.
18. P e ñ a-A y a l a, A. Educational Data Mining: A Survey and a Data Mining-Based Analysis of Recent Works. – Expert Systems with Applications, Vol. **41**, 2014, No 4, pp. 1432-1462.
19. Q u i n l a n, J. R. Simplifying Decision Trees. – International Journal of Human-Computer Studies, Vol. **51**, 1999, No 2, pp. 497-510.
20. R o m e r o, C., S. V e n t u r a. Educational Data Mining: A Survey from 1995 to 2005. – Expert Systems with Application, Vol. **33**, 2007, No 1, pp. 135-146.
21. R o m e r o, C., S. V e n t u r a. Educational Data Mining: A Review of the State of the Art. – IEEE Transactions on Systems, Man and Cybernetics, Vol. **40**, 2010, No 6, pp. 601-618.
22. S e n, B., E. U c a r, D. D e l e n. Predicting and Analyzing Secondary Education Placement Test Scores: A Data Mining Approach. – Expert Systems with Applications, Vol. **39**, 2012, No 10, pp. 9468-9476.
23. S h a h i r i, A. M., W. H u s a i n, N. A. R a s h i d. A Review on Predicting Students' Performance Using Data Mining Techniques. – Procedia Computer Science, Vol. **72**, 2015, pp. 414-422.
24. S t r e c h t, P., J. M e n d e s-M o r e i r a, C. S o a r e s. Merging Decision Trees: A Case Study in Predicting Student Performance. – In: Advanced Data Mining and Applications. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 535-548.
25. S t r e c h t, P., L. C r u z, C. S o a r e s, J. M e n d e s-M o r e i r a, R. A b r e u. A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance. – In: Proc. of 8th International Conference on Educational Data Mining (EDM'15), 2015, pp. 392-395.
26. T h a i-N g h e, N., P. J a n e c e k, P. H a d d a w y. A Comparative Analysis of Techniques for Predicting Academic Performance. – In: Proc. of 37th Annual Frontiers in Education Conference – Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007, pp. T2G-7–T2G-12.
27. T h a i-N g h e, N. Predicting Student Performance in an Intelligent Tutoring System. PhD Thesis at Hildesheim University, 2011.

28. T h a i-N g h e, N., T. H o r v a t h. Personalized Forecasting Student Performance. – In: Proc. of 11th IEEE International Conference on Advanced Learning Technologies (ICALT'11), 2011, pp. 412-414.
29. T o s c h e r, A., M. J a h r e r. Collaborative Filtering Applied to Educational Data Mining. KDD Cup 2010: Improving Cognitive Models with Educational Data Mining, 2010.
30. Z i m m e r m a n n, J., K. H. B r o d e r s e n, J.-P. P e l l e t, E. A u g u s t, J. M. B u h m a n n. Predicting Graduate-Level Performance from Undergraduate Achievements. – In: Proc of 4th International Educational Data Mining Conference (EDM'11), 2011, pp. 357-358.