Grade Prediction via Prior Grades and Text Mining on Course Descriptions: Course Outlines and Intended Learning Outcomes

Jiawei Li  
Nanyang Technological University  
jiawei009@e.ntu.edu.sg

S. Supraja  
Nanyang Technological University  
supraja.s@ntu.edu.sg

Wei Qiu  
Nanyang Technological University  
qiuwei@ntu.edu.sg

Andy W. H. Khong  
Nanyang Technological University  
andykhong@ntu.edu.sg

ABSTRACT

Academic grades in assessments are predicted to determine if a student is at risk of failing a course. Sequential models or graph neural networks that have been employed for grade prediction do not consider relationships between course descriptions. We propose the use of text mining to extract semantic, syntactic, and frequency-based features from course content. In addition, we classify intended learning outcomes according to their higher- or lower-order thinking skills. A learning parameter is then formulated to model the impact of these cognitive levels (that are expected for each course) on student performance. These features are then embedded and represented as graphs. Past academic achievements are then fused with the above features for grade prediction. We validate the performance of the above approach via datasets corresponding to three engineering departments collected from a university. Results obtained highlight that the proposed technique generates meaningful feature representations and outperforms existing methods for grade prediction.

Keywords

Grade prediction, graph networks, course descriptions, semantic similarities, cognitive levels

1. INTRODUCTION

Detecting students at the risk of failing university courses based on predicted grades is essential for administering early intervention strategies. From a regression problem perspective, grades obtained from prior courses in previous semesters are used to predict grades for pilot courses registered in the upcoming semester.


© 2022 Copyright is held by the author(s). This work is distributed under the Creative Commons Attribution NonCommercial NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license.

https://doi.org/10.5281/zenodo.6853171

1.1 Related Models for Grade Prediction

Existing techniques for grade prediction using past academic records include conventional regression models such as random forest, support vector machine, and K-nearest neighbor [1, 10, 15] as well as the factorization machine in a collaborative filtering setting [33]. In addition to the use of past examination results, information derived from online click-stream data on learning management systems has been used to augment the prediction capability of a model [25, 26]. More recently, sequential models such as the long short-term memory (LSTM) have been developed to capture the temporal dynamics of past academic performance [12]. While such deep learning models have achieved reasonable success in grade prediction, existing temporal-based approaches do not take the relationships among courses and among students into account. Consideration of these relationships is essential since information pertaining to courses with similar content and students with similar cognitive levels would aid in grade prediction. In addition, the performance trend of an academically-inclined student or a well-performed course in the current semester may continue for the upcoming semesters [23].

Notwithstanding the above, graph neural networks have recently been employed to generate meaningful feature representations which model the transitions of grade distributions between courses across semesters [11]. Similar to social multi-relational networks [14] with nodes representing either students or courses, three graphs—student-course, student-student, and course-course graphs—consisting of edge links computed via grade distribution similarities or correlations have been constructed [21, 23]. Modeling the student-course relations have also been achieved via knowledge graphs to extract course and student embeddings as well as to encode temporal student behavioral data [17]. Pre- or co-requisites between courses have also been considered for grade prediction [27].

Despite adopting multi-dimensional approaches toward analyzing prior course grades to predict student performance [35], existing models assume that the relationship among courses depends solely on the grade distribution; these models do not consider topics covered and the intended
learning outcomes defined by the course instructors. These aspects are important since the process of knowledge acquisition often involves assimilating and discerning information from myriad sources [29], i.e., academic performance has shown to be dependent on prior experience and how the student has understood certain concepts. Moreover, course content that overlap or are highly inter-dependent may influence how well the student can achieve the intended learning outcomes for the upcoming semesters [38]. While course syllabus has recently been used to extract frequency-based features for the determination of course similarities [16], it does not analyze the intended learning outcomes nor capture the relationship between courses holistically. It is also not surprising to expect that students who are less academically inclined often struggle in courses that require higher-order thinking skills. Information pertaining to the thinking skills required for prior courses will, therefore, allow the grade-prediction model to better represent grades achieved from previous semesters.

1.2 Grade Prediction From Curriculum Development Perspective

From a curriculum development perspective, course descriptions comprise topics to be covered and the intended learning outcomes for each course designed by the course instructor [34]. The importance of identifying suitable topics is motivated by an earlier study where first-year university students who had been exposed to fundamental concepts in high school have shown to perform better than those who had not studied similar content before [13]. In today’s context, this highlights the intrinsic (and often intimate) relationships including pre-requisites, recommended literature, and course content that define dependencies between courses. Coupled with the fact that course instructors often adopt the constructivist approach in curriculum design [6], analysis of course content is important for grade prediction.

Apart from course content, outcome-based teaching and learning require course instructors to identify suitable intended learning outcomes and assessments that measure those learning outcomes [4,30]. In this regard, learning activities with various cognitive complexity levels should be designed and aligned with the learning outcomes constructively throughout the course [2,5,9,32]. Alignment of learning activities can be achieved via the revised Bloom’s Taxonomy with the recollection of information being associated with the lowest-order thinking skill to generating creative outcomes being associated with the highest-order thinking skill [20]. Given that less academically-inclined students often face challenges in higher-order thinking skills [39], it is important to consider the influence of learning outcomes on student performance for grade prediction.

1.3 Contribution of This Work

In this work, we propose a course description-based grade prediction (CODE-GP) model that employs text mining techniques for extracting features associated with (i) course content similarities and (ii) higher- or lower-order thinking skills required for each course. With regards to the first dimension highlighted in Table 1, we propose three types of course similarities extracted from topic outlines and intended learning outcomes found within course descriptions. These similarities include semantic [22], syntactic [3], and frequency-based features [36]. The use of these features is in contrast to the use of grade distributions as edge weights for generating similarities [11]. The basis for our proposed architecture is motivated by the need to consider both course outlines and intended learning outcomes, since both the intended learning outcomes and syllabus are important for the development and implementation of teaching programs [28]. In addition, we also consider past performance of each student from the perspective of thinking skills required for each course. In particular, the proposed model employs a document classification approach that tags each course with higher- or lower-order thinking skills according to the revised Bloom’s Taxonomy. A learnable parameter is then used to aggregate the respective grades achieved for both lower- and higher-order thinking skill courses. This allows the proposed model to establish the relationship between the complexity of courses and academic performance.

As shown in Figure 1, we adopt graph neural networks to generate representations of the above text mining features. These features are represented as course- and student-similarity graphs with nodes corresponding to courses and students, respectively. The edge weights for the former are computed based on the proposed three text features. For the latter, past academic grades are aggregated, and the similarity related to Jensen-Shannon Divergence (JSD) is then computed among the grade distributions [23]. These graphs are subsequently embedded and trained using a graph convolutional network (GCN) layer.

In addition and similar to [12], we incorporate temporal information extracted from past examination records for each student across semesters. Grade embeddings, the corresponding student vector, and prior course vectors acquired from the GCN for each semester are then concatenated as a representation vector. This temporal representation serves as the input to LSTM, which exploits the sequential relationships and predicts the grade for a course to be taken in the coming semester.

Table 1: Overview of text mining approaches in the proposed CODE-GP model

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course similarities</td>
<td>Semantic</td>
<td>Contextual closeness of course content</td>
</tr>
<tr>
<td>Course similarities</td>
<td>Syntactic</td>
<td>Grammatical differences across cognitive levels</td>
</tr>
<tr>
<td>Frequency-based</td>
<td></td>
<td>Overlapping works appearance in descriptions</td>
</tr>
<tr>
<td>Student similarities</td>
<td>Higher-order</td>
<td>Verbs corresponding to creative outcomes</td>
</tr>
<tr>
<td>Lower-order</td>
<td></td>
<td>Verbs corresponding to recalling concepts</td>
</tr>
</tbody>
</table>

In addition and similar to [12], we incorporate temporal information extracted from past examination records for each student across semesters. Grade embeddings, the corresponding student vector, and prior course vectors acquired from the GCN for each semester are then concatenated as a representation vector. This temporal representation serves as the input to LSTM, which exploits the sequential relationships and predicts the grade for a course to be taken in the coming semester.
2. THE PROPOSED CODE-GP MODEL

The task of grade prediction involves predicting the grade for student $s_i$ who has registered a pilot course. Given $N_C$ number of prior courses and $N_S$ students, the set of prior courses is defined as $C = \{c_1, c_2, \ldots, c_{N_C}\}$ and the set of students as $S = \{s_1, s_2, \ldots, s_{N_S}\}$. We define $\tilde{g}_i$, as the predicted grade of a given pilot course for the student $s_i$.

2.1 Construction of Course Similarities

Graph Based on Course Descriptions

The CODE-GP model incorporates semantic, syntactic, and frequency-based features extracted from course descriptions that comprise topic outlines and intended learning outcomes. These features are subsequently used for constructing the course-similarity graph. We first pre-process the text by removing symbols, diagrams, equations, numbers, punctuation marks, and stop-words (e.g., “and”, “or”). All remaining characters are set to lower case [32].

Semantic similarity based on word embeddings has been employed to assess student capability for the recommendation of similar courses [24]. In the context of CODE-GP, we first define topic outline as $q_i$ corresponding to course $c_i$. A topic outline vector $v_{q_i}$ is then generated from $q_i$ based on the bidirectional encoder representations from transformer (BERT) embeddings [7, 8]. The cosine similarity between $q_i$ and $q_j$ is then computed between two course outlines via

$$\cos(\theta(q_i, q_j)) = \frac{v_{q_i} \cdot v_{q_j}}{||v_{q_i}|| ||v_{q_j}||}.$$  \hspace{1cm} (1)

With $0 \leq \cos(\theta(q_i, q_j)) \leq 1$, a value of 1 implies an almost semantically similar pair of courses $c_i$ and $c_j$.

Syntactic features for CODE-GP comprise phrase types (i.e., regular expressions (regexes)) that are extracted from statements associated with the intended learning outcomes. In this context, we first extract noun- and verb-phrases from the intended learning outcome document $l_i$ corresponding to course $c_i$. These (multiple) phrases are then associated with their parts-of-speech tags resulting in the set of regexes $R_i$ [31]. Overlaps between the regex sets are then computed via the Jaccard similarity given by

$$\Phi(c_i, c_j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|},$$  \hspace{1cm} (2)

where $0 \leq \Phi(c_i, c_j) \leq 1$. The number of common occurrences is denoted by $|R_i \cap R_j|$ while $|R_i \cup R_j|$ refers to the total number of regexes. A high Jaccard similarity, therefore, implies a high proportion of similar phrase types occurring between a course pair regardless of whether the topics covered are identical.

The term frequency-inverse document frequency (TF-IDF) determines the uniqueness of a word within a set of documents [37]. To account for word appearance similarity, we include TF-IDF weighting on both the topic outlines $q_i$ and intended learning outcomes $l_i$ for course $c_i$. These features are extracted from each concatenated course document $d_i = q_i \cup l_i$, where $\cup$ denotes the concatenation of two texts. We then compute the cosine similarity $\Omega(c_i, c_j) \propto v_{d_i} \cdot v_{d_j}$ similar to (1) but between bag of words (BoW) vectors $v_{d_i}$ and $v_{d_j}$ corresponding to each course document. Here, $v_{d_i} = [\alpha(w_1, d_i), \alpha(w_2, d_i), \ldots, \alpha(w_{NW}, d_i)]$ with $w_k$ denoting the $k$th word in document $d_i$. The BoW vector length is based on the word vocabulary size $NW$ across the entire corpus. The value of each element corresponding to the TF-IDF weight for word $w_k$ is given by [37]

$$\alpha(w_k, d_i) = \frac{N_{w_k,d_i}}{L(d_i)} \times \log\left(\frac{N_D}{N_{w_k} + 1}\right).$$  \hspace{1cm} (3)
where $N_{w_k d_i}$ is the number of times $w_k$ occurs in $d_i$, $L(d_i)$ denotes the length of that document, $N_D$ the total number of documents, and $N_{w_i}$ the number of documents in which $w_i$ occurs. The obtained TF-IDF values are subsequently normalized to prevent bias in the term frequency variable due to document length $L(d_i)$.

With nodes of the course-similarity graph denoted by each course $c_i \in C$, the edge weights are determined via

$$a_{ij} \propto \left( \beta_{\text{semantic}} \times \cos(\theta(c_i, c_j)) \right),$$

$$\beta_{\text{frequency}} \times \Omega(c_i, c_j),$$

where $\beta_{\text{semantic}}, \beta_{\text{syntactic}}$, and $\beta_{\text{frequency}}$ are the trainable weights. Each of the variable $a_{ij}^c$ is used within the adjacency matrix

$$A_c = \begin{bmatrix}
a_{i1} & \cdots & a_{iN_C} \\
\vdots & \ddots & \vdots \\
a_{N_C i} & \cdots & a_{N_C N_C}
\end{bmatrix}$$

(5)

corresponding to the course-similarity graph.

### 2.2 Temporal Grade Information

Before attempting the pilot course in the current semester, we assume, for each student $s_i$, availability of prior course grades in $C$ across semesters $t \in \{1, \ldots, N_T\}$, where $N_T$ is the total number of semesters. $g_{si}^t$ denotes the grade that student achieves for $c_i$ in semester $t$. Hence, the grade vector for student $s_i$ in semester $t$ is given by

$$g_{si}^t = [g_{si1}^t, \ldots, g_{si N_C}^t],$$

(6)

where $N_c$ is the total number of prior courses across all $N_T$ semesters. It is important to note that for a given semester, only a subset of these $N_c$ prior courses are attempted, i.e., $g_{si}^t$ is not a full vector and null elements will be assigned for courses not attempted during that semester. Across all previous $N_T$ semesters, we acquire the temporal grade information for each student, as shown in Figure 1. Such temporal grade information would be used in two ways—(i) being aggregated according to the thinking skills required for each course and to generate student similarity as will be described in Section 2.3 and (ii) being concatenated with the course and student embeddings as input for LSTM.

### 2.3 Construction of Student Similarities

#### Graph Based on Cognitive Levels

Construction of the student-similarity graph is based on cognitive levels associated with each course according to Table 1. Each of the prior courses is first categorized as one that requires high-order thinking skills $H$ or lower-order thinking skills $L$. This is achieved by first classifying each course intended learning outcome statement via document classification described in [31] with classes being defined according to Bloom’s Taxonomy. Each course is then tagged as $H$ (or $L$) if more statements are classified as labels associated with high-order (or lower-order) thinking skills.

For each student, we compute the frequency distribution $p_{c_i}^s$ and $p_{c_i}^p$ corresponding to courses that require lower- and higher-order thinking skills. This is achieved by first dividing the grade range (1-100) into five bins of twenty-point intervals before determining the number of courses (in each $H$ and $L$ category) that falls under each bin. Contributions of these two distributions are then learned via

$$p_{si} = \beta_g \times p_{c_i}^p + (1 - \beta_g) \times p_{c_i}^H,$$

(7)

where $\beta_g$ is a learnable weight for $p_{si}$. With the above, student similarities are obtained via the JSD between the grade distribution for each pair of students, i.e.,

$$a_{si}^g = 1 - \text{JSD}(p_{si} || p_{si}).$$

(8)

Therefore, a higher $a_{si}^g$ implies that the two students possess similar higher- or lower-order skills (measured by how they perform in the prior courses). With the student similarity graph shown in Figure 1 comprising students as nodes, the corresponding adjacency matrix $A_s$ is generated based on $a_{si}^g$ similar to (5).

### 2.4 GCN and Embeddings

After constructing the course- and student-similarity graphs, we employ a two-layer GCN to embed each graph. Both course and student nodes are encoded with one-hot vectors to obtain encoded matrices $X_C$ and $X_S$. The embedding vector $E_C$ for the course-similarity graph is generated via

$$E_C = W_C X_C$$

(9)

such that the one-hot vectors are represented as dense vectors of lower dimensions. Here, $W_C$ is the weight matrix. With $E_S$ being generated similarly, and with $A_C$ and $A_S$ derived from Sections 2.1 and 2.3, two GCN layers [18] are then applied to obtain latent representations of all nodes in course-similarity graph $C$ and student-similarity graph $S$. In particular, the $(G + 1)$th layer for $C$ is computed via

$$Z_c^{(G+1)} = \sigma \left( D_c^{-\frac{1}{2}} A_C D_c^{-\frac{1}{2}} Z_c^{(g)} W_c^{(g)} \right),$$

(10)

where $D_c = \sum_i A_c$ is the degree matrix, $Z_c^{(0)} = E_C$, and $W_c^{(g)}$ is the weight matrix. The output of the GCN for course-similarity graph is denoted as matrix $R_C = Z_c^{(2)}$ with each row vector $r_{ci}$ being associated with course $c_i$. The above computation is also applied on student-similarity graph $S$ to obtain the graph embedding matrix $R_S$ with each row vector being defined as $r_{si}$ for student $s_i$.

To generate representations for the prior grades achieved, embedding is applied for each grade. With a one-hot vector representing a unique value of prior grade $g_{si1}$, the embedding vector for a student prior grade is learned via

$$e_{si1,c_j} = W_G \text{ One-hot} \left( g_{si1,c_j} \right),$$

(11)
where $W_G$ is the weight matrix. The three embedding vectors from course-similarity graph, student-similarity graph, and temporal grade information are then concatenated for each semester to form a $(l_a + N_C \times (l_b + l_c)) \times 1$ vector

$$e_i^s = [r_{s_j}, r_{s_j}, e_{s_i,c_1}, \ldots, r_{s_{N_C}}, e_{s_i,c_{N_C}}]^T,$$

where $l_a$, $l_b$, and $l_c$ denote the embedding length for $r_{s_j}$, $r_{c_j}$, and $e_{s_i,c_j}$, respectively, and $T$ denotes transpose. Each of these vectors are then concatenated to form a feature matrix

$$E_i = [e_i^1, \ldots, e_i^{N_T}]$$

of each student $s_i$ for the subsequent prediction model.

### 2.5 Grade Prediction using LSTM

LSTM models time-series representations and is used to predict the pilot grade based on sequential matrix $E_i$, for each student. Through the use of input, output, and forget gate, LSTM aggregates important and permutes less significant representations to achieve prediction of pilot grades in semester $N_T + 1$. LSTM is employed for grade prediction via the hidden state

$$h_i^t = \text{LSTM}(e_i^t, h_i^{t-1}),$$

where $h_i^t$ denotes the hidden state for semester $t$. The predicted grade $\hat{g}_i$ for student $s_i$ obtained from the last hidden state is then given by

$$\hat{g}_i = w_L \cdot h_i^{N_T} + b,$$

where $w_L$ and $b$ are defined, respectively, as the weight vector and bias scalar for the predictor.

## 3. RESULTS AND DISCUSSION

### 3.1 Datasets and Implementation Details

Open-source datasets employed for grade prediction do not include course descriptions. We collected data that include both academic records and course descriptions (comprising both course outlines and intended learning outcomes). These are obtained from three engineering departments in a university to evaluate the models. Each dataset is obtained with the student name and identity being hashed by another office (authorized to handle such data) to protect privacy. Table 2 summarizes details for each dataset used. In particular, $N_T$ for each dataset is determined by the maximum number of semesters the students within the cohort take to complete all courses under consideration. The prior course list for each pilot course consists of the core courses corresponding to the department’s curriculum. In addition, $N_C$ and $N_S$ are distinct for each dataset. In our experiments, the training, validation, and testing ratio are set as 6:2:2.

We employed the mean squared error (MSE)

$$MSE = \frac{1}{N_S} \sum_{i=1}^{N_S} (\hat{g}_i - g_i)^2$$

for performance evaluation, where $g_i$ denotes the actual grade obtained by student $s_i$ for a given pilot course. In terms of hyperparameter selection, course description document embeddings are trained using BERT with a dimension of 768. During GCN training, the dropout rate was set as 0.5, while the Adam optimizer with a learning rate of 0.001 was used. A weight decay parameter was set to $5 \times 10^{-4}$ to prevent overfitting.
3.2 Performance Analysis

We take pilot course $c_3$ from Department 3 as an example to illustrate the impact of considering the semantic, syntactic, and frequency aspects of words used in course outlines and intended learning outcomes. Three heatmaps with colors depicting the similarity values described in Section 2.2 are provided while details pertaining to prior course information are shown in Table 3.

Figure 2(a) illustrates the semantic cosine similarity where high similarities in terms of course content are indicated by the dark shades. It can be seen that the mathematics-based prior course EC180 exhibits high semantic similarity with other prior courses EC280, EA206, EC181, and EA304, which have high mathematical content. On the other hand, computing course CS108 exhibits lower semantic similarity with the most of other non-programming courses. Figure 2(b) highlights how (dis)similar phrase types are between the course outlines and the intended learning outcomes of two prior courses. We note that EC181 exhibits higher Jaccard similarity with courses that require fundamental scientific and mathematical knowledge such as EC180, 1C102, and EC280. TF-IDF weighting, on the other hand, indicates the choice and uniqueness of words being used in the course outlines and intended learning outcomes. Figure 2(c) highlights the high variability in words used between the courses being considered—only a few pairs of course outlines and intended learning outcomes exhibit high TF-IDF similarity. In addition, we also note that the similarity between content is irrelevant. This can be observed from the fact that even though EC180 and EC181 are mathematics-related, their frequency-based TF-IDF similarity is relatively low.

We next compare the performance of the proposed CODE-GP model with LSTM based grade-prediction model [12], GCN [19], and the conventional logistic regression (LR) model. While LR and LSTM focus on temporal information and GCN exploits the interrelationship between courses and students, the proposed model considers both aspects. We note from Table 4 that the proposed CODE-GP model achieves the highest grade prediction capability than the LR, LSTM, and GCN. While the proposed model requires higher complexity than these three baseline models, CODE-GP achieves the lowest mean MSE of 0.0222 (11.9% improvement compared to LSTM), across the three departments as seen in Table 4. These results highlight the importance of course descriptions when constructing student- and course-similarity graphs with time series information. Features extracted from course descriptions enhance the grade prediction capability instead of using only a single modality.

We further performed an ablation test by excluding each input graph/temporal representations. Table 5 summarizes the MSE and mean absolute error (MAE) across all three departments. We note that the use of all three aspects in CODE-GP is vital to provide a holistic perspective for grade prediction. It is interesting to note that grade prediction performance is more sensitive to course-similarity graph (compared to student-similarity graph). This suggests that information derived from course descriptions can assist in grade prediction since performance is closely related to achieving the set of intended learning outcomes depicted in course descriptions. These results also highlight that temporal information and graphs provide complementary features which contribute jointly to the success of grade prediction.

4. CONCLUSIONS

We propose a grade prediction model that considers course descriptions and prior academic results. Text mining techniques determine the edge weights of the course- and student-similarity graphs. A three-pronged model that constitutes the semantic, syntactic, and frequency-based feature extraction methods is formulated for course similarities. Student performance in terms of their achievements in courses associated with low- or high-order thinking skills have also been incorporated to construct the student-similarity graph. The LSTM synthesizes these aspects before performing prediction.

An accurate and just-in-time prediction of performance enables course instructors to administer early interventions. Once the predicted results indicate a tendency of a student in failing a course, student support staff can respond and plan for a personalized intervention strategy for each student. Moreover, early detection of at-risk students can potentially reduce the drop-out rate. Future work may include techniques that incorporate other data modalities such as student demographic or online learning behavior while protecting student privacy.

5. REFERENCES


[34] R. Tang and W. Sae-Lim. Data science programs in U.S. higher education: An exploratory content


