Will this Course Increase or Decrease Your GPA? Towards Grade-aware Course Recommendation

Sara Morsy
and Engineering
University of Minnesota
morsy@cs.umn.edu

George Karypis
and Engineering
University of Minnesota
karypis@cs.umn.edu

In order to help undergraduate students towards successfully completing their degrees, developing tools that can assist students during the course selection process is a significant task in the education domain. The optimal set of courses for each student should include courses that help him/her graduate in a timely fashion and for which he/she is well-prepared for so as to get a good grade in. To this end, we propose two different *grade-aware course recommendation* approaches to recommend to each student his/her optimal set of courses. The first approach ranks the courses by using an objective function that differentiates between courses that are expected to increase or decrease a student's GPA. The second approach combines the grades predicted by grade prediction methods with the rankings produced by course recommendation methods to improve the final course rankings. To obtain the course rankings in both approaches, we adapt two widely-used representation learning techniques to learn the optimal temporal ordering between courses. Our experiments on a large dataset obtained from the University of Minnesota that includes students from 23 different majors show that the grade-aware course recommendation methods can do better on recommending more courses in which the students are expected to perform well and recommending fewer courses which they are expected not to perform well in than grade-unaware course recommendation methods.

Keywords: course recommendation, grade prediction, SVD, course2vec, representation learning, GPA

1. Introduction

The average six-year graduation rate across four-year higher education institutions has been around 59% over the past 15 years (Kena et al., 2016; Braxton et al., 2011), while less than half of college graduates finish within four years (Braxton et al., 2011). These statistics pose challenges in terms of workforce development, economic activity and national productivity. This has resulted in a critical need for analyzing the available data about past students in order to provide actionable insights to improve college student graduation and retention rates. Some examples of the problems that have been investigated are: course recommendation (Elbadrawy and Karypis, 2016; Bendakir and Aïmeur, 2006; Lee and Cho, 2011; Parameswaran and Garcia-Molina, 2009; Parameswaran et al., 2010; Parameswaran et al., 2010; Parameswaran et al., 2011), next-term course grade prediction (Polyzou and Karypis, 2016; Sweeney et al., 2016;

Elbadrawy and Karypis, 2016; Morsy and Karypis, 2017; Hu and Rangwala, 2018), predicting the final grade of the course based on the student's ongoing performance during the term (Meier et al., 2015), in-class activities grade prediction (Elbadrawy et al., 2015), predicting student's performance in tutoring systems (Thai-Nghe et al., 2011; Hershkovitz et al., 2013; Hwang and Su, 2015; Romero et al., 2008; Thai-Nghe et al., 2012), and knowledge tracing and student modeling (Reddy et al., 2016; Lan et al., 2014; González-Brenes and Mostow, 2012).

Both *course recommendation* (Bendakir and Aïmeur, 2006; Parameswaran et al., 2011; Elbadrawy and Karypis, 2016; Bhumichitr et al., 2017; Hagemann et al., 2018) and *grade prediction* (Sweeney et al., 2016; Elbadrawy and Karypis, 2016; Polyzou and Karypis, 2016; Morsy and Karypis, 2017; Hu and Rangwala, 2018) methods aim to help students during the process of course registration in each semester. By learning from historical registration data, course recommendation focuses on recommending courses to students that will help them in completing their degrees. Grade prediction focuses on estimating the students' expected grades in future courses. Based on what courses they previously took and how well they performed in them, the predicted grades give an estimation of how well students are prepared for future courses. Nearly all of the previous studies have focused on solving each problem separately, though both problems are inter-related in the sense that they both aim to help students graduate in a timely and successful manner.

In this paper, we propose a *grade-aware course recommendation* framework that focuses on recommending a set of courses that will help students: (i) complete their degrees in a timely fashion, and (ii) maintain or improve their GPA. To this end, we propose two different approaches for recommendation. The first approach ranks the courses by using an objective function that differentiates between courses that are expected to increase or decrease a student's GPA. The second approach uses the grades that students are expected to obtain in future courses to improve the ranking of the courses produced by course recommendation methods.

To obtain course rankings in both approaches, we adapt two widely-known representation learning techniques, which have proven successful in many fields, to solve the grade-aware course recommendation problem. The first is based on singular value decomposition, which is a linear model that learns a low-rank approximation of a given matrix. The second, which we refer to as Course2vec, is based on word2vec (Mikolov et al., 2013) that uses a log-linear model to formulate the problem as a maximum likelihood estimation problem. In both approaches, the courses taken by each student are treated as temporally-ordered sets of courses, and each approach is trained to learn these orderings.

1.1. Contributions

The main contributions of this work are the following:

1. We propose a *Grade-aware Course Recommendation* framework in higher education that recommends courses to students that the students are most likely to register for in their following terms and that will help maintain or improve their overall GPA. The proposed framework combines the benefits of both course recommendation and grade prediction approaches to better help students graduate in a timely and successful manner.

2. We investigate two different approaches for solving grade-aware course recommendation. The first approach uses an objective function that explicitly differentiates between good and bad courses, while the other approach combines grade prediction methods with course recommendation methods in a non-linear way.

3. We adapt two-widely used representation learning techniques to solve the grade-aware course recommendation problem, by modeling historical course ordering data and differentiating between courses that increase or decrease the student's GPA.

4. We perform an extensive set of experiments on a dataset spanning 16 years obtained from the University of Minnesota, which includes students who belong to 23 different majors. The results show that: (i) the proposed grade-aware course recommendation approaches outperform grade-unaware course recommendation methods in recommending more courses that increase the students' GPA and fewer courses that decrease it; and (ii) the proposed representation learning approaches outperform competing approaches for grade-aware course recommendation in terms of recommending courses which students are expected to perform well in, as well as differentiating between courses which students are expected to perform well in and those which they are expected not to perform well in.

5. We provide an in-depth analysis of the recommendation accuracy across different majors and different student groups. We show the effectiveness of our proposed approaches on different majors and student groups over the best competing method. In addition, we analyze two important characteristics for the recommendations: the course difficulty as well as the course popularity. We show that our proposed approaches are not prone to recommending easy courses. Furthermore, they are able to recommend courses with different popularity in a similar manner.

2. Related Work

2.1. Course Recommendation

Different machine learning methods have been recently developed for course recommendation. For example, Bendakir and Aïmeur (2006) used association rule mining to discover significant rules that associate academic courses from previous students' data. Lee and Cho (2011) ranked the courses for each student based on the course's importance within his/her major, its prerequisites, and the extent by which the course adds to the student's knowledge state.

Another set of recommendation methods proposed in (Parameswaran and Garcia-Molina, 2009; Parameswaran et al., 2010; Parameswaran et al., 2010; Parameswaran et al., 2011) focused on satisfying the degree plan's requirements that include various complex constraints. The problem was shown to be NP-hard and different heuristic approaches were proposed in order to solve the problem.

Elbadrawy and Karypis (2016) proposed using both student- and course-based academic features, in order to improve the performance of three popular recommendation methods in the education domain, namely popularity-based ranking, user-based collaborative filtering and matrix factorization. These features are used to define finer groups of students and courses and were shown to improve the recommendation performance of the three aforementioned methods than using coarser groups of students.

The group popularity ranking method proposed by Elbadrawy and Karypis (2016) (**grp-pop**), ranks the courses based on how frequently they were taken by students of the same major and academic level as the target student. Though this is a simple ranking method, it was shown to be among the best performing methods proposed by the authors. This is due to the domain restrictions, where each degree program offers a specific set of required and elective courses for the students to choose a subset from, and a pre-requisite structure exists among most of these courses.

Pardos et al. (2019) proposed a similar course2vec model that was done in parallel to our proposed work[1]. They used a skip-gram neural network architecture that takes as input one course and outputs multiple probability distributions over the courses. The approaches that are presented here differ from that work because their model is grade-unaware, while ours is grade-aware, which is a principal contribution of our work.

Another model (Backenköhler et al., 2018) that is also parallel and most relevant to our work also proposed to combine grade prediction with course recommendation. Our work is different in two aspects. First, Backenköhler et al. (2018) use a course dependency graph constructed using the Mann-Whitney U-test as the course recommendation method. This graph consists of nodes that represent courses and directed edges between them. A directed edge going from course A to course B means that the chance of getting a better grade in B is higher when A is taken before B than when A is not taken before B. One limitation of this approach is that, for pairs (A, B) of courses that do not have sufficient data about A not being taken before B, no directed edge will exist from A to B, despite the fact that there may be sufficient data about A followed by B, which may imply that A is a pre-requisite for B. Our proposed representation learning approaches for course recommendation, described in Section 3.1., on the other hand, are able to learn all possible orderings for pairs of courses that have sufficient data. In addition, the course embeddings are learned in a way such that courses taken after a common set of courses are located close in the latent space, which enables discovering new relationships between previous and subsequent courses that do not necessarily exist in the data.

Second, we propose a new additional approach for grade-aware course recommendation, which modifies the course recommendation objective function to differentiate between good and bad sequences of courses and does not require a grade prediction method.

2.2. Course Sequence Discovery and Recommendation

Though our focus in this paper is to recommend courses for students in their following term, and not to recommend the whole sequence of courses for all terms, our proposed models try to learn the sequencing of courses such that they predict the next good courses based on the previously-taken set of courses.

Cucuringu et al. (2017) utilized several ranking algorithms, e.g., PageRank, to extract a global ranking of the courses, where the rank here denotes the order in which the courses are taken by students. The discovered course sequences were used to infer the hidden dependencies, i.e., informal prerequisites, between the courses, and to understand how/if course sequences learned from high- and low-performing students are different from each other. This technique learns only one global ranking of courses from all students, which cannot be used for personalized recommendation.

Xu et al. (2016) proposed a course sequence recommendation framework that aimed to minimize the time-to-graduate, which is based on satisfying pre-requisite requirements, course availability during the term, the maximum number of courses that can be taken during each term, and degree requirements. They also proposed to do joint optimization of both graduation time and GPA by clustering students based on some contextual information, e.g., their high school rank and SAT scores, and keeping track of each student's sequence of taken courses as well as his/her

[1] An earlier version of our paper was published as a technical report at https://goo.gl/HrxVdr.

GPA. Then, for a new student, he/she is assigned to a specific cluster based on their contextual information and the sequence of courses from that cluster that has the highest GPA estimate is recommended to him/her. This framework can work well on the more restricted degree programs that have little variability between the degree plans taken by students, given that there is enough support for the different degree plans from past students. However, the more flexible degree programs have much variability in the degree plans taken by their students, as shown in Morsy and Karypis (2019). This makes an exact extraction system like the one above inapplicable for their students, unless there exists a huge dataset that covers the many different possible sequences with high support.

2.3. Representation Learning

Representation learning has been an invaluable approach in machine learning and artificial intelligence for learning from different types of data such as text and graphs. Objects can be represented in a vector space via local or distributed representations. Under local (or one-hot) representations, each object is represented by a binary vector of size equal to the total number of objects, where only one of the values in the vector is one and all the others are set to zero. Under distributed representations, each object is represented by a vector, which can come from hand-engineered features that are usually sparse and high-dimensional, or a learned representation, called "embeddings" in a latent space that preserves the relationships between the objects, which is usually low-dimensional and more practical than the former.

A widely used approach for learning object embeddings is Singular Value Decomposition (SVD) (Golub and Reinsch, 1970). SVD is a traditional low-rank approximation method that has been used in many fields. In recommendation systems, a user-item rating matrix is typically decomposed into the user and item latent factors that recover the observed ratings in the matrix, e.g., (Sarwar et al., 2000; Bell et al., 2007; Paterek, 2007; Koren, 2008).

Recently, neural networks have gained a lot of interest for learning object embeddings in different fields for their ability to handle more complex relationships than SVD. Some of the early well-known architectures include word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014), which were proposed for learning distributed representations for words. For instance, neural language models for words, phrases and documents in Natural Language Processing (Huang et al., 2012; Mikolov et al., 2013; Le and Mikolov, 2014; Pennington et al., 2014; Mikolov et al., 2013) are now widely used for different tasks, such as machine translation and sentiment analysis. Similarly, learning embeddings for graphs, such as DeepWalk (Perozzi et al., 2014), LINE (Tang et al., 2015) and node2vec (Grover and Leskovec, 2016) were shown to have performed well on different applications, such as multi-label classification and link prediction. Moreover, learning embeddings for products in e-commerce and music playlists in cloud-based music services has been recently proposed for next basket recommendation (Chen et al., 2012; Grbovic et al., 2015; Wang et al., 2015).

3. Grade-aware Course Recommendation

Undergraduate students often achieve inconsistent grades in the various courses they take, which may increase or decrease their overall GPA. This is illustrated in Figure 1, which shows the histogram of differences between each grade obtained by a student over his/her prior average grade for the dataset used in our experiments (Table 1). As we can see, more than 10% of the

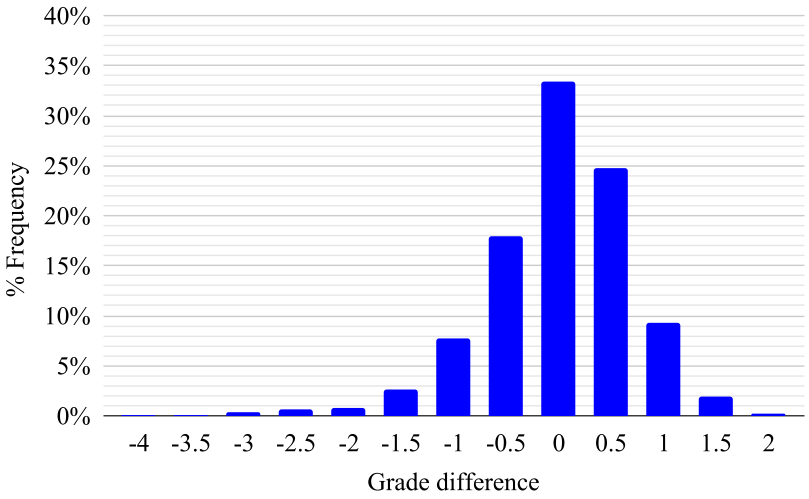


Figure 1: Grade difference from the student's average previous grade.

grades are a full letter grade lower than the corresponding students' previous average grades[2]. The poor performance in some of these courses can result in students having to retake the same courses at a later time or to increase the number of courses that they will have to take in order to graduate with a desired GPA. As a result, this will increase the financial cost associated with obtaining a degree and can incur an opportunity cost by delaying the students' graduation.

For the cases in which a student's performance in a course is a result of him/her not being well-prepared for it (i.e., is taking the course at the wrong time in his/her studies), course recommendation methods can be used to recommend a set of courses for that student that will help (i) him/her in completing his/her degree in a timely fashion and (ii) maintain or improve his/her GPA. We will refer to the methods that do those simultaneously as **grade-aware course recommendation** approaches. Note that the majority of the existing approaches cannot be used to solve this problem as they ignore the performance the student is expected to get in the courses that they recommend.

In this work, we propose two different approaches for grade-aware course recommendation. The first approach (Section 3.1.) uses two representation learning approaches that explicitly differentiate between courses in which the student is expected to perform well and courses in which the student is expected not to perform well. The second approach (Section 3.2.) combines grade prediction methods with course recommendation methods to improve the final course rankings. The goal of both approaches is to rank the courses in which the student is expected to perform well higher than those in which he/she is expected not to perform well.

3.1. Grade-aware Representation Learning Approaches

Our first approach for solving the grade-aware course recommendation problem relies on modifying the way we use the previous students' data to differentiate between courses in which the student is expected to perform well and courses in which the student is expected not to perform

[2] The letter grading system in this dataset has 11 letter grades (A, A-, B+, B, B-, C+, C, C-, D+, D, F) that correspond to the numerical grades (4, 3.67, 3.33, 3, 2.67, 2.33, 2, 1.67, 1.33, 1, 0), with A being the highest grade and F the lowest one.

well. As such, for every student, we define a course taken by him/her to be **a good (subsequent) course** if the student's grade in it is equal to or higher than his/her average previous grade, otherwise, we define that course to be **a bad (subsequent) course**. The goal of our method is to recommend to each student a set of good courses.

Motivated by the success of representation learning approaches in recommendation systems (Koren, 2008; Chen et al., 2012; Grbovic et al., 2015; Wang et al., 2015), we adapt two widely-used approaches to solve the grade-aware course recommendation problem. The first approach applies the SVD linear factorization model on a co-occurrence frequency matrix that differentiates between good and bad courses (Section 3.1.1.), while the second one optimizes an objective function of a neural network log-linear model that differentiates between good and bad courses (Section 3.1.2.).

In both approaches, the courses taken by each student are treated as temporally-ordered sets of courses, and each approach is trained on these data in order to learn the proper ordering of courses as taken by students. The course representations learned by these models are then used to create personalized rankings of courses for students that are designed to include courses that are relevant to the students' degree programs and will help them maintain or increase their GPAs.

3.1.1. Singular Value Decomposition

SVD (Golub and Reinsch, 1970) is a traditional low-rank linear model that has been used in many fields. It factorizes a given matrix $\mathbf{X}$ by finding a solution to $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, where the columns of $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular vectors, respectively, and $\boldsymbol{\Sigma}$ is a diagonal matrix containing the singular values of $\mathbf{X}$. The d largest singular values, and corresponding singular vectors from $\mathbf{U}$ and $\mathbf{V}$, is the rank d approximation of $\mathbf{X}$ ($\mathbf{X}_d = \mathbf{U}_d\boldsymbol{\Sigma}_d\mathbf{V}_d^T$). This technique is called truncated SVD.

Since we are interested in learning course ordering as taken by past students, we apply SVD on a previous-subsequent co-occurrence frequency matrix $\mathbf{F}$, where F_{ij} is the number of students in the training data that have previously taken course i before they took course j.

We form two different previous-subsequent co-occurrence frequency matrices, as follows. Let n_{ij}^{+} and n_{ij}^{-} be the number of students who have taken course i before course j, where course j is considered a good course for the first group and a bad course for the second one, respectively. The two matrices are:

1. $\mathbf{F}^{+}$: where $F_{ij}^{+} = n_{ij}^{+}$.
2. $\mathbf{F}^{+-}$: where $F_{ij}^{+-} = n_{ij}^{+} - n_{ij}^{-}$.

We scaled the rows of each matrix to $L1$ norm and then applied truncated SVD on them. The course embeddings are then given by $\mathbf{U}_d\sqrt{\boldsymbol{\Sigma}_d}$ and $\mathbf{V}_d\sqrt{\boldsymbol{\Sigma}_d}$ for the previous and subsequent courses, respectively.

Note that we append a (+), or (+-) as a superscript to the matrix and as a suffix to the corresponding method's name based on what course information it utilizes during learning and how it utilizes it. A (+)-based method utilizes the good course information only and ignores the bad ones, while a (+-)-based method utilizes both the good and bad course information and differentiates between them.

Recommendation. Given the previous and subsequent course embeddings estimated by SVD, course recommendation is performed as follows. Given a student s with his/her previously-taken set of courses, $c_1, \ldots, c_k$, who would like to register for his/her following term, we compute his/her implicit profile by averaging over the embeddings of the courses taken by him/her in all previous terms[3]. We then compute the dot product between s's profile and the embeddings of each candidate course $c_t \in C$. Then, we rank the courses in non-increasing order according to these dot products, and select the top courses as the final recommendations for s.

3.1.2. Course2vec

The above SVD model works on pairwise, one-to-one relationships between previous and subsequent courses. We also model course ordering using a many-to-one, log-linear model, which is motivated by the recent word2vec Continuous Bag-Of-Word (CBOW) model (Mikolov et al., 2013). Word2vec works on sequences of individual words in a given text, where a set of nearby (context) words (i.e., words within a pre-defined window size) are used to predict the target word. In our case, the sequences would be the ordered terms taken by each student, where each term contains a set of courses, and the previous set of courses would be used to predict future courses for each student.

MODEL ARCHITECTURE. We formulate the problem as a maximum likelihood estimation problem. Let $\mathcal{T}_i = \{c_1, \ldots, c_n\}$ be a set of courses taken in some term i. A sequence $Q_s = \langle \mathcal{T}_1, \ldots, \mathcal{T}_m \rangle$ is an ordered list of m terms as taken by some student s, where each term can contain one or more courses. Let $\mathbf{W} \in \mathbb{R}^{|C| \times d}$ be the courses' representations when they are treated as *previous* courses, and let $\mathbf{W'} \in \mathbb{R}^{d \times |C|}$ be their representations when they are treated as "subsequent" courses, where $|C|$ is the number of courses and d is the number of dimensions in the embedding space. We define the probability of observing a future course c_t given a set of previously-taken courses $c_1, \ldots, c_k$ using the softmax function, i.e.,

$$Pr(c_t | c_1, \ldots, c_k) = y_t = \frac{\exp(\mathbf{w'}_{c_t}^{T} \mathbf{h})}{\sum_{j=1}^{C} \exp(\mathbf{w'}_{c_j}^{T} \mathbf{h})}, \tag{1}$$

where $\mathbf{h}$ denotes the aggregated vector of the representations of the previous courses, where we use the average pooling for aggregation, i.e.,

$$\mathbf{h} = \frac{1}{k} \mathbf{W}^T (\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_k),$$

where $\mathbf{x}_i$ is a one-hot encoded vector of size $|C|$ that has 1 in the c_i's position and 0 otherwise. The Architecture for Course2vec is shown in Figure 2. Note that one may consider more complex neural network architectures, which is left for future work.

We propose the two following models:

1. **Course2vec(+)**. This model maximizes the log-likelihood of observing only the good subsequent courses that are taken by student s in some term given his/her previously-taken

[3] We tried using different window sizes for the number of previous terms. Using all previous terms achieved better results than using one, two or three previous terms only.

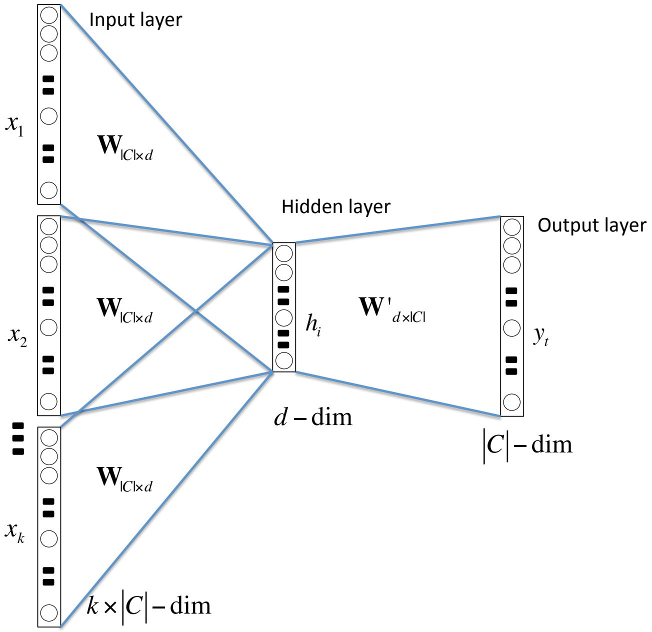


Figure 2: Neural network architecture for Course2vec.

set of courses. The objective function of Course2vec(+) is thus:

$$\underset{\mathbf{W},\mathbf{W}'}{\text{maximize}} \sum_{s\in\mathcal{S}} \sum_{\mathcal{T}_i\in Q_s} \Big(\log Pr(\mathcal{G}_{s,i}|\mathcal{P}_{s,i}) \Big), \tag{2}$$

where: $\mathcal{S}$ is the set of students, $\mathcal{G}_{s,i}$ is the set of good courses taken by student s at term i, and $\mathcal{P}_{s,i}$ is the set of courses taken by student s prior to term i. Note that i starts from 2, since the previous set of courses $\mathcal{P}_{s,i}$ would be empty for $i = 1$.

2. **Course2vec(+-)**. This model maximizes the log-likelihood of observing good courses and minimizes the log-likelihood of observing bad courses given the set of previously-taken courses. The objective function of Course2vec(+-) is thus:

$$\begin{aligned} \underset{\mathbf{W},\mathbf{W}'}{\text{maximize}} \textstyle\sum_{s\in\mathcal{S}} \sum_{\mathcal{T}_i\in Q_s} \Big(\quad & \log Pr(\mathcal{G}_{s,i}|\mathcal{P}_{s,i}) \\ & -\log Pr(\mathcal{B}_{s,i}|\mathcal{P}_{s,i}) \Big), \end{aligned} \tag{3}$$

where: $\mathcal{B}_{s,i}$ is the set of bad courses taken by student s at term i, and the rest of the terms are as defined in Eq. 2.

Note that Course2vec(+) is analogous to SVD(+) and Course2vec(+-) is analogous to SVD-(+-) in terms of how they utilize the good and bad courses in the training set.

MODEL OPTIMIZATION. The objective functions in Eqs. 2 and 3 can be solved using Stochastic Gradient Descent (SGD), by solving for one subsequent course at a time. The computation of gradients in the two equations requires computing Eq. 1 for all courses for the denominator, which requires knowing whether a course is to be considered a good or a bad subsequent course for a given context. However, not all the relationships between every context (previous set of courses) and every subsequent course are known from the data. Hence, for each context, we only update the subsequent course vector when the course is known to be a good or bad subsequent course associated with that context. In the case that some context does not have a sufficient pre-defined number of subsequent courses with known relationships, then we randomly sample a few other courses and treat them as bad courses, similar to the negative sampling approach used in word2vec (Le and Mikolov, 2014).

Note that in Course2vec(+-), since a course can be seen as both a good and a bad subsequent course for the same context in the data (for different students), then, in this case, we randomly choose whether to treat that course as good or bad each time according to a uniform distribution that is based on its good and bad frequency in the dataset. In addition, for both Course2vec(+) and Course2vec(+-), if the frequency between a context and a subsequent course is less than a pre-defined threshold, e.g., 20, then we randomly choose whether to update that subsequent course's vector in the denominator each time it is visited. The code for Course2vec can be found at https://zenodo.org/record/3464635#.XZFTEJNKjNY, which is built on the original word2vec code that was implemented for the CBOW model[4].

Recommendation Given the previous and subsequent course embeddings estimated by Course2vec, course recommendation is performed as follows. Given a student s with his/her previously-taken set of courses, $c_1, \ldots, c_k$, who would like to register for his/her following term, we compute the probability $Pr(c_t|c_1, \ldots, c_k)$ for each candidate course $c_t \in C$ according to Eq. 1. We then rank the courses in non-increasing order according to their probabilities and select the top courses as the final recommendations for s. Note that since the denominator in Eq. 1 is the same for all candidate courses, the ranking score for course c_t can be simplified to the dot product between $\mathbf{w'}_{c_t}$ and $\mathbf{h}$, where $\mathbf{h}$ represents the student's implicit profile.

3.2. Combining Course Recommendation with Grade Prediction

The second approach that we developed for solving the grade-aware course recommendation problem relies on using the grades that students are expected to obtain in future courses to improve the ranking of the courses produced by course recommendation methods. Our underlying hypothesis behind this approach is that a course that both is ranked high by a course recommendation method and has a high predicted grade should be ranked higher than one that either has a lower ranking by the recommendation method or is predicted to have a lower grade in it. This in turn will help improve the final course rankings for students by taking both scores into account simultaneously.

Let $\hat{g}_{s,c}$ be the predicted grade for course c as generated from some grade prediction model, and let $\hat{r}_{s,c}$ be the ranking score for c as generated from some course recommendation method. We combine both scores to compute the final ranking score for c as follows:

$$\text{rank-score}_{s,c} = \hat{g}_{s,c}^{\alpha} \times (|\hat{r}_{s,c}|)^{(1-\alpha)} \times \text{sign}(\hat{r}_{s,c}), \tag{4}$$

where α is a hyper-parameter in the range $(0, 1)$ that controls the relative contribution of $\hat{g}_{s,c}$ and $\hat{r}_{s,c}$ to the overall ranking score, and sign($\hat{r}_{s,c}$) denotes the sign of $\hat{r}_{s,c}$, i.e., 1 if $\hat{r}_{s,c}$ is positive and -1 otherwise. Note that both $\hat{g}_{s,c}$ and $\hat{r}_{s,c}$ are standardized to have zero mean and unit variance.

[4] Original code is at: https://goo.gl/UvUuMQ

In this work, we will use the representation learning approaches described in Section 3.1. as the course recommendation method. We will also use the grade-unaware variations of each of them (see Section 4.2.) to compare combining the grade prediction methods with both recommendation approaches.

To obtain the grade prediction score, we will use *Cumulative Knowledge-based Regression Models* (CKRM) (Morsy and Karypis, 2017). CKRM is a set of grade prediction methods that learn low-dimensional as well as textual-based representations for courses that denote the required and provided knowledge components for each course. It represents a student's knowledge state as the sum of the provided knowledge component vectors of the courses taken by them, weighted by their grades in them. CKRM then predicts the student's grade in a future course as the dot product between their knowledge state vector and the course's required knowledge component vector. We will denote the recommendation method that combines CKRM with SVD and Course2vec as **CKRM+SVD** and **CKRM+Course2vec**, respectively.

4. Experimental Evaluation

4.1. Dataset Description and Preprocessing

The data used in our experiments was obtained from the University of Minnesota and spans a period of 16 years (Fall 2002 to Summer 2017). From that dataset, we extracted the degree programs that have at least 500 graduated students through Fall 2012, which accounted for 23 different majors from different colleges. For each of these degree programs, we extracted all the students who graduated from this program and extracted the 50 most frequent courses taken by the students as well as the courses that belonged to frequent subjects, e.g., CSCI is a subject that belongs to the Computer Science department at the University. A subject is considered to be frequent if the average number of courses that belong to that subject over all students is at least three. This filtering was made to remove the courses we believe are not relevant to the degree program of students. We also removed any courses that were taken as pass/fail.

Using the above dataset, we split it into train, validation and test sets as follows. All courses taken before Spring 2013 were used for training, courses taken between Spring 2013 and Summer 2014 inclusive were used for validation, and courses taken afterwards (Fall 2014 to Summer 2017 inclusive) were used for test purposes.

At the University of Minnesota, the letter grading system has 11 letter grades (A, A-, B+, B, B-, C+, C, C-, D+, D, F) that correspond to numerical grades (4, 3.667, 3.333, 3, 2.667, 2.333, 2, 1.667, 1.333, 1, 0). For each (context, subsequent) pair in the training, validation, and test set, where the context represents the previously-taken set of courses by a student, the context contained only the courses taken by the student with grades higher than the D+ letter grade, which the student does not have to repeat. The statistics of the 23 degree programs are shown in Table 1.

4.2. Baseline and Competing Methods

We compare the performance of the proposed representation learning approaches against competing approaches for grade-aware course recommendation, which are defined as follows:

- **Grp-pop(+-)**: We modified the group popularity ranking method developed in Elbadrawy and Karypis (2016) and explained in Section 2. for grade-aware course recommendation.

Table 1: Dataset statistics.

Major	# Students	# Courses	# Grades
Accounting (ACCT)	661	55	7,614
Aerospace Engr. (AEM)	866	72	13,280
Biology (BIOL)	1,927	113	15,590
Biology, Soc. & Envir. (BSE)	1,231	56	9,389
Biomedical Engr. (BME)	1,002	64	13,808
Chemical Engr. (CHEN)	1,045	82	10,219
Chemistry (CHEM)	765	78	7,814
Civil Engr. (CIVE)	1,160	74	15,992
Communication Studies (COMM)	2,547	90	17,135
Computer Science & Engr. (CSE)	1,790	98	13,520
Electrical Engr. (ECE)	1,197	84	12,781
Elementary Education (ELEM)	1,283	60	15,303
English (ENGL)	1,790	113	12,451
Finance (FIN)	1,326	55	12,150
Genetics, Cell Biol. & Devel. (GCD)	843	92	9,726
Journalism (JOUR)	2,043	91	23,549
Kinesiology (KIN)	1,499	161	23,451
Marketing (MKTG)	2,077	51	13,084
Mechanical Engr. (MECH)	1,501	79	25,608
Nursing (NURS)	1,501	88	18,239
Nutrition (NUTR)	940	71	12,400
Political Science (POL)	1,855	111	13,904
Psychology (PSY)	3,047	100	25,299

For each course c, let n_c^+ and n_c^- be the number of students that have the same major and academic level as the target student s, where c was considered a good subsequent course for the first group and a bad one for the second group. We can differentiate between good and bad subsequent courses using the following ranking score (which is similar to the (+-)-based approaches):

$$\text{rank-score}_{s,c} = n_c^+ - n_c^-. \tag{5}$$

- **Grp-pop(+)**: Here, the group popularity ranking method considers only the good subsequent courses, similar to SVD(+) and Course2vec(+). Specifically, the ranking score is computed as

  $$\text{rank-score}_c = n_c^+,$$

  where n_c^+ is as defined in Eq. 5.

- **Course dependency graph**: This is the course recommendation method utilized in (Backenköhler et al., 2018) (see Section 2.1.).

We also compare the performance of the representation learning approaches for both grade-aware and grade-unaware course recommendation. The grade-unaware representation learning approaches are defined as follows:

- **SVD(++)**: Here, SVD is applied on the previous-subsequent co-occurrence frequency matrix: $\mathbf{F}^{++}$: where $F_{ij}^{++} = n_{ij}^+ + n_{ij}^-$.

- **Course2vec(++)**. This model maximizes the log-likelihood of observing all courses taken by student s in some term given the set of previously-taken courses, regardless of the subsequent course being a good or a bad one. This can be written as:

  $$\underset{\mathbf{W},\mathbf{W}'}{\text{maximize}} \sum_{s \in \mathcal{S}} \sum_{\mathcal{T}_i \in Q_s} \Big(\log Pr(\mathcal{C}_{s,i} | \mathcal{P}_{s,i}) \Big),$$

where: $\mathcal{C}_{s,i}$ is the set of courses taken by student s at term i, and the rest of the terms are as defined in Eq. 2.

Note that, here we append a (++) suffix to the grade-unaware variation of the method's name since it utilizes all the course information without differentiating between good and bad courses.

4.3. Evaluation Methodology and Metrics

Previous course recommendation methods used the recall metric to evaluate the performance of their methods. The goal of the proposed grade-aware course recommendation methods is to recommend to the student courses which he/she is expected to perform well in and not recommend courses which he/she is expected not to perform well in. As a result, we cannot use the recall metric as is, and instead, we use three variations of it that differentiate between good and bad courses. The first, *Recall(good)*, measures the fraction of the actual good courses that are retrieved. The second, *Recall(bad)*, measures the fraction of the actual bad courses that are retrieved. The third, *Recall(diff)*, measures the overall performance of the recommendation method in ranking the good courses higher than the bad ones.

The first two metrics are computed as the average of the student-term-specific corresponding recalls. In particular, for a student s and a target term t, the first two recall metrics for that (s, t) tuple are computed as:

1. Recall(good)$_{(s,t)} = \frac{\left|G_{s,n_{(s,t)}}\right|}{n^g_{(s,t)}}$.

2. Recall(bad)$_{(s,t)} = \frac{\left|B_{s,n_{(s,t)}}\right|}{n^b_{(s,t)}}$.

$G_{s,n_{(s,t)}}$ and $B_{s,n_{(s,t)}}$ denote the set of good and bad courses, respectively, that were taken by s in t and exist in his/her list of $n_{(s,t)}$ recommended courses, $n_{(s,t)}$ is the actual number of courses taken by s in t, and $n^g_{(s,t)}$ and $n^b_{(s,t)}$ are the actual number of good and bad courses taken by s in t, respectively. Since our goal is to recommend good courses only, we consider a method to perform well when it achieves a high Recall(good) and a low Recall(bad).

Recall(diff) is computed as the difference between Recall(good) and Recall(bad), i.e.,

3. Recall(diff) = Recall(good) - Recall(bad).

Recall(diff) is thus a signed measure that assesses both the degree and direction to which a recommendation method is able to rank the actual good courses higher than the bad ones in its recommended list of courses for each student, so the higher the Recall(diff) value, the better the recommendation method is.

To further analyze the differences in the ranking results of the proposed approaches, we also computed the following two metrics:

- **Percentage GPA increase/decrease:** Let cur-good$_s$ and cur-bad$_s$ be the current GPA achieved by student s on the good and bad courses recommended by some recommendation method, respectively, and let prev-gpa$_s$ be his/her GPA prior to that term. Then, the percentage GPA increase and decrease are computed as:

$$\%\text{ GPA increase} = \frac{\text{cur-good}_s - \text{prev-gpa}_s}{\text{prev-gpa}_s} \times 100.0.$$

$$\% \text{ GPA decrease} = \frac{\text{prev-gpa}_s - \text{cur-bad}_s}{\text{prev-gpa}_s} \times 100.0.$$

- **Coverage for good/bad terms:** The number of terms where some recommendation method recommends at least one good (or bad) subsequent course will be referred to as its coverage for good (or bad) terms. The higher the coverage for good terms by some method, the more students will get good recommendations that will maintain or improve their overall GPA. On the other hand, the lower the coverage for bad terms, the less students will get bad recommendations that will decrease their overall GPA.

We compute the above two metrics for the terms on which the recommendation method recommends at least one of the actual courses taken in that term. For each method, the percentage GPA increase and decrease as well as the coverage for good and bad terms are computed as the average of the individual scores. Since we would like to recommend courses that optimize the student's GPA, the higher the GPA percentage increase and the coverage for good terms and the lower the GPA percentage decrease and the coverage for bad terms that a method achieves, the better the method is.

Note that a recommendation is only done for students who have taken at least three previous courses. For each (s, t) tuple, the recommended list of courses using any method are selected from the list of courses that are being offered at term t only, and that were not already taken by s with an associated grade that is either: (i) $\geq$ C+, or, (ii) $\geq \mu_s - 1.0$, where μ_s is the average previous grade achieved by s. Therefore, we only allow recommending repeated courses in the case that the student has achieved a low grade in it such that the course's credits do not add to the earned credits, or when they a achieve bad grade in them relative to their grades in previous terms. This filtering technique significantly improved the performance of all the baseline and proposed methods.

4.4. Model Selection

We did an extensive search in the parameter space for model selection. The parameters in the SVD-based models is the number of latent dimensions (d). The parameters in the Course2vec-based models are the number of latent dimensions (d), and the minimum number of subsequent courses ($samples$), in the denominator of Eq. 1 that are used during the SGD process of learning the model. We experimented with the parameter d in the range $[10 - 30]$ with a step of 5, with the minimum number of $samples$ with the values $\{3, 5\}$, and with the parameter α in Eq. 4 in the range $[0.1 - 0.9]$ with a step of 0.2.

For each major, the training set was used for learning the distributed representations of the courses, whereas the validation set was used to select the best performing parameters in terms of the highest Recall(diff).

5. Results

We evaluate the effectiveness of the proposed grade-aware course recommendation methods in order to answer the following questions:

RQ1. How do the SVD- and Course2vec-based approaches for course recommendation compare to each other?

Table 2: Prediction performance of the proposed representation learning-based approaches for grade-aware course recommendation.

Metric	SVD		Course2vec	
	(+)	(+-)	(+)	(+-)
Recall(good)	0.468	0.396	0.448	0.351
Recall(bad)	0.372	0.206	0.404	0.202
Recall(diff)	0.096	0.190	0.044	0.149

Note: Underlined entries indicate best performance.

RQ2. How do the combination of grade prediction with representation learning approaches compare to each other?

RQ3. How do the two proposed approaches for solving grade-aware course recommendation compare to each other?

RQ4. How do the proposed approaches compare to competing approaches for grade-aware course recommendation?

RQ5. What are the benefits of grade-aware course recommendation over grade-unaware course recommendation?

RQ6. How does the recommendation accuracy vary across different majors and student subgroups?

RQ7. What are the characteristics of the recommended courses, in terms of course difficulty and popularity?

5.1. Comparison of the Representation Learning Approaches for Grade-aware Course Recommendation

Table 2 shows the prediction performance of the two proposed representation learning approaches for grade-aware course recommendation. The results show that SVD(+) achieves the best Recall(good), while SVD(+-) achieves the best Recall(diff). Course2vec(+-) achieves the best Recall(bad), which is comparable to SVD(+-).

By comparing the corresponding SVD and Course2vec approaches, we see that SVD outperforms Course2vec in almost all cases. We believe this is caused by the fact that there is a limited number of positive training data for Course2vec since only the good courses are used as positive examples for learning the models. This is supported by the comparable prediction performance of the (++)-based approaches that use all the available training data as positive examples, which are shown in Table 5.

By comparing the (+)- and (+-)-based methods, we see that the (+-)-based model achieves a worse Recall(good), but a much better Recall(bad). For instance, SVD(+-) achieves a 15% decrease in Recall(good) and a 45% decrease in Recall(bad) over SVD(+). This is expected since utilizing the bad course information gives the models more power to learn to rank these courses low, but it also adds some noise, since different students with the same or similar previous set of courses can achieve different outcomes on the same courses.

Table 3: Prediction performance of combining CKRM with the representation learning-based approaches for grade-aware course recommendation methods.

Metric	CKRM + SVD			CKRM + Course2vec		
	(++)	(+)	(+-)	(++)	(+)	(+-)
Recall(good)	0.438	0.417	0.385	0.411	0.417	0.338
Recall(bad)	0.279	0.230	0.189	0.269	0.264	0.183
Recall(diff)	0.158	0.187	0.197	0.142	0.152	0.155

Note: Underlined entries indicate best performance.

5.2. Comparison of the Grade-aware Recommendation Approaches Combining Grade Prediction with Course Recommendation

Table 3 shows the prediction performance of the grade-aware recommendation approaches that combine CKRM with the grade-aware and grade-unaware representation learning methods. The results show that CKRM+SVD(++) achieves the best Recall(good), while CKRM+Course2vec-(+-) achieves the best Recall(bad). Overall, CKRM+SVD(+-) achieves the best Recall(diff). Combining CKRM with the grade-unaware, i.e., (++)-based, approaches helped in differentiating between good and bad courses, by achieving a high Recall(diff) of 0.158 and 0.142 for SVD and Course2vec, respectively. However, despite these performance improvements, the combinations that use the grade-aware recommendation methods do better. For instance, CKRM+SVD(+) outperforms CKRM+SVD(++) by 15% in terms of Recall(diff).

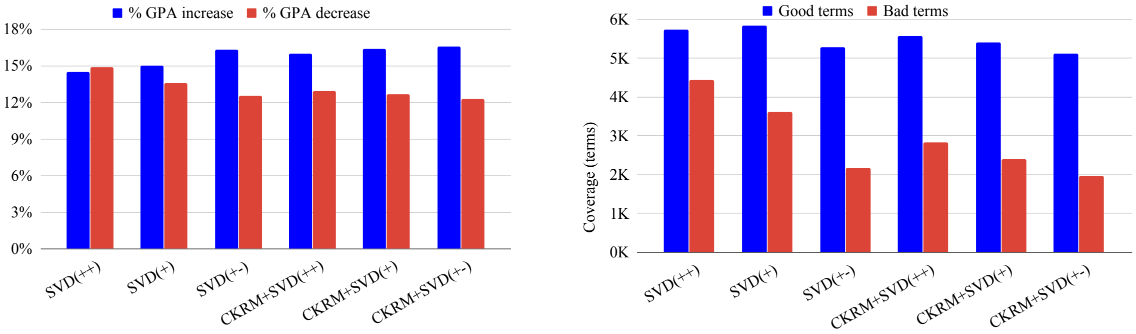
The results also show that the SVD-based (+)- and (+-)-based approaches outperform their Course2vec counterparts in terms of Recall(diff), similar to the results of SVD and Course2vec alone (Section 5.1.). Unlike the difference in the performance of SVD(+) vs SVD(+-), CKRM-+SVD(+) achieves a similar Recall(diff) to that achieved by CKRM+SVD(+-) (and the same holds for the Course2vec-based approaches). The difference is that CKRM+SVD(+) achieves higher Recall(good) and Recall(bad) than CKRM+SVD(+-).

5.3. Comparison of the Proposed Approaches for Grade-aware Course Recommendation

Comparing each of the SVD- and Course2vec-based approaches with and without CKRM (shown in Tables 2 and 3), we see that combining CKRM with the (+)-based approaches improved their performance with 95% and 245% increase in Recall(diff) for SVD and Course2vec, respectively. On the other hand, combining CKRM with the (+-)-based approaches achieves comparable performance to using the corresponding (+-)-based approach alone.

By further analyzing these ranking results, Figure 3 shows the percentage GPA increase and decrease as well as the coverage for good and bad terms for each SVD-based method with and without CKRM[5]. CKRM+SVD(+) outperforms SVD(+) in all but one metric, which is coverage for good terms, where it achieves slightly worse performance than SVD(+). On the other hand, CKRM+SVD(+-) has comparable performance to SVD(+-), which is analogous to their recall metrics results.

[5] The results of the Course2vec-based methods are similar, and are thus omitted.



(a) Percentage GPA increase and decrease

(b) Coverage for good and bad terms

Figure 3: Performance of the different SVD-based methods with and without CKRM (refer to Section 4.3. for the metrics definitions).

5.4. Representation Learning vs Competing Approaches for Grade-aware Course Recommendation

Table 4 shows the prediction performance of the representation learning and competing approaches for grade-aware course recommendation. Grp-pop(+-) achieves the best Recall(diff) among the three competing (baseline) approaches. The results also show that SVD(+) achieves the best Recall(good), while grp-pop(+-) achieves the best Recall(bad). Overall, SVD(+-) achieves the best Recall(diff).

5.5. Grade-aware vs Grade-unaware Representation Learning Approaches

Table 5 shows the performance prediction of the representation learning approaches for grade-aware, i.e., (+)- and (+-)-based approaches, and grade-unaware, i.e., (++)-based approach, course recommendation. Each of SVD(+) and Course2vec(+) achieves a Recall(good) that is comparable to or better than that achieved by its corresponding (++)-based approach. In addition, both the (+)- and (+-)-based methods achieve much better (lower) Recall(bad). For instance, SVD(+) and SVD(+-) achieve 0.372 and 0.206 Recall(bad), respectively, resulting in 26% and 59% improvement over SVD(++), respectively.

By comparing the (++)-, (+)-, and (+-)-based approaches in terms of Recall(diff), we can see that the (++)-based approaches achieve negative recall values which indicate that they recommend more bad courses than they recommend good ones. The (+)-based approaches do slightly

Table 4: Prediction performance of the representation learning-based vs competing approaches for grade-aware course recommendation.

Metric	Dependency Graph	Grp-pop (+)	Grp-pop (+-)	SVD (+)	SVD (+-)	Course2vec (+)	Course2vec (+-)
Recall(good)	0.382	0.425	0.367	0.468	0.396	0.448	0.351
Recall(bad)	0.260	0.343	0.188	0.372	0.206	0.404	0.202
Recall(diff)	0.122	0.082	0.179	0.096	0.190	0.044	0.149

Note: Underlined entries indicate best performance.

Table 5: Prediction performance of the representation learning based approaches for grade-aware and grade-unaware course recommendation.

Metric	SVD (++)	Course2vec (++)	SVD (+)	Course2vec (+)	SVD (+-)	Course2vec (+-)
Recall(good)	0.453	0.455	0.468	0.448	0.396	0.351
Recall(bad)	0.502	0.493	0.372	0.404	0.206	0.202
Recall(diff)	-0.048	-0.038	0.096	0.044	0.190	0.149

Note: Underlined entries indicate best performance.

better, while the (+-)-based approaches achieve the highest Recall(diff). This is expected since the (++)-based methods treat both types of subsequent courses equally during their learning, and so they recommend both types in an equal manner. This shows that differentiating between good and bad courses in any course recommendation method is very helpful for ranking the good courses higher than the bad ones, which will help the student maintain or improve their overall GPA.

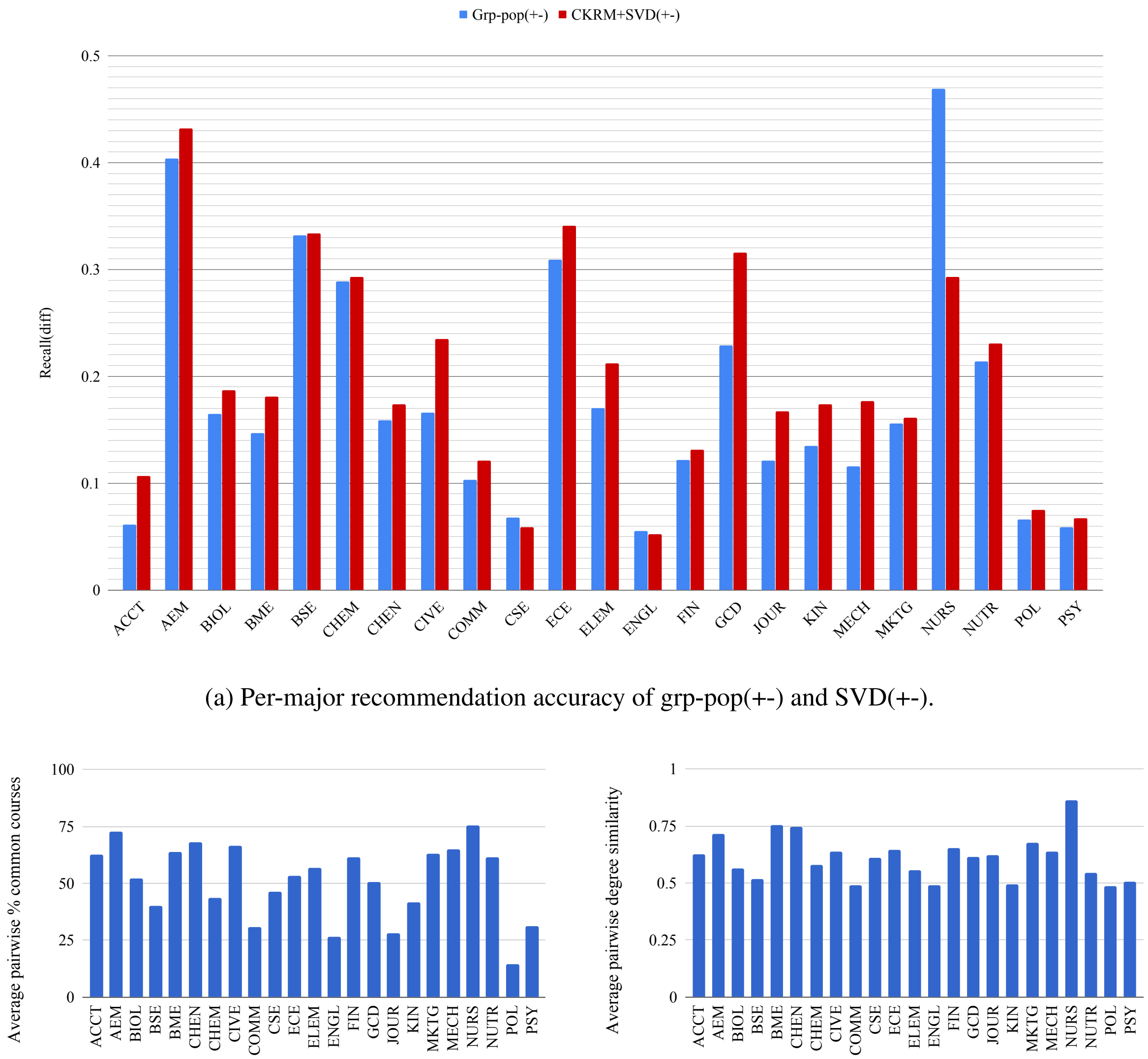
In terms of percentage GPA increase and decrease (shown in Figure 3), SVD(+-) outperforms SVD(++) by 2% in percentage GPA increase and 2.5% in percentage GPA decrease. Moreover, SVD(+-) achieves $\sim 62\%$ less coverage for the bad terms than SVD(++), while it achieves $\sim 10\%$ less coverage for the good terms.

5.6. Analysis of Recommendation Accuracy

Our discussion so far focused on analyzing the performance of the different methods by looking at metrics that are aggregated across the different majors. However, given that the structure of the degree programs of different majors is sometimes quite different, and that different student groups can exhibit different characteristics, an important question that arises is how the different methods perform across the individual degree programs and different student groups and if there are methods that consistently perform well across majors as well as across student groups. In this section, we analyze the recommendations done by one of our best performing models, CKRM+SVD(+-), against the best performing baseline, i.e., grp-pop(+-), in terms of Recall(diff), across these degree programs and student groups (RQ6).

5.6.1. Analysis of Different Majors

Table 4 shows the recommendation accuracy, in terms of Recall(diff), across the 23 majors, by both grp-pop(+-) and CKRM+SVD(+-) (Fig 4a). First, we can see that there is a huge variation in the recall values across the majors, ranging from 0.05 to $\sim$0.5. Second, we see that CKRM+SVD(+-) consistently outperforms grp-pop(+-), except for the nursing major. To further look into why this happens, we investigated some of the characteristics of the students' degree sequences. For each major, we computed the pairwise percentage of common courses among students who belong to that major, which is shown in Figure 4b. In addition, we computed the similarity in the sequencing, i.e., ordering, of the common courses between each pair of students, which is shown in Figure 4c. For computing the pairwise degree similarity, we utilized the formula proposed in (Morsy and Karypis, 2019), which computes the degree similarity



(a) Per-major recommendation accuracy of grp-pop(+-) and SVD(+-).

(b) Pairwise % common courses per major.

(c) Pairwise degree similarity per major.

Figure 4: Per-major recommendation accuracy and the characteristics of the students' degrees.

between a pair of degree plans d_1 and d_2 as:

$$\text{sim}(d_1, d_2) = \frac{\sum_{(x,y)\in|C_1 \cap C_2|} T(t_{1,x} - t_{1,y}, t_{2,x} - t_{2,y})}{|C_1 \cap C_2|}, \tag{6}$$

where C_i is the set of courses taken in degree i, and $t_{i,x}$ is the time, i.e., term number, that course x was taken in d_i, e.g., the first term is numbered 1, the second is numbered 2 and so forth. Function $T(dt_1, dt_2)$ is defined as:

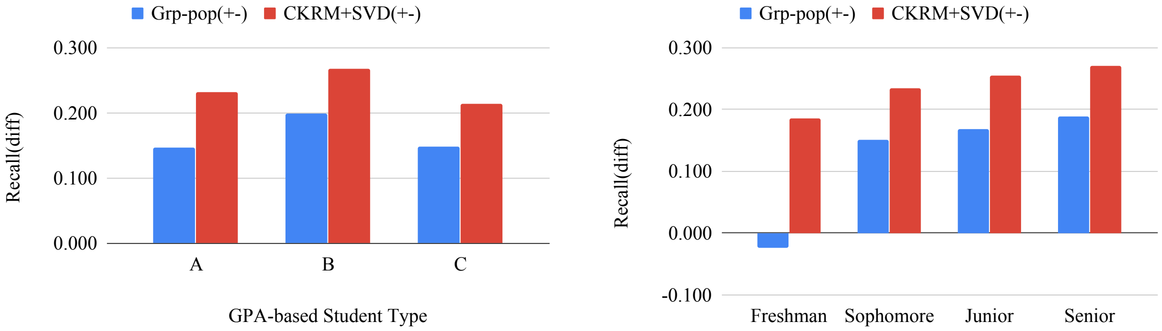
$$T(dt_1, dt_2) = \begin{cases} 1, & \text{if } dt_1 = dt_2 = 0 \\ \exp\Big(-\lambda(|dt_1 - dt_2|)\Big), & \text{if } dt_1 \times dt_2 \geq 1 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where λ is an exponential decay constant. Function T assigns a value of 1 for pairs of courses taken concurrently, i.e., during the same term, in both plans, and assigns a value of 0 for pairs of courses that are either: (i) taken in reversed order in both plans, or (ii) taken concurrently in one plan and sequentially in the other. For pairs of courses taken in the same order, it assigns a positive value that decays exponentially with $|dt_1 - dt_2|$.

We found that there is a high correlation between the Recall(diff) values and both the average pairwise percentage of common courses and the average pairwise degree similarity among students of these majors (correlation values of 0.47 and 0.5 for grp-pop(+-), and 0.47 and 0.38 for CKRM+SVD(+-), respectively). This implies that as the percentage of common courses and degree similarity between pairs of students decrease, accurate course recommendation becomes more difficult since there is more variability in the set of courses taken as well as their sequencing. The nursing major, where grp-pop(+-) outperforms CKRM+SVD(+-) has the highest average pairwise percentage of common courses, $\sim$76%, as well as the highest average pairwise degree similarity, $\sim$0.86, compared to all other majors. This implies that the nursing major is the most restricted major and that students tend to follow highly similar degree plans and take very similar courses at each academic level. The group popularity ranking in this case can easily outperform other recommendation methods.

5.6.2. Analysis of Different Student Groups

Figure 5 shows the recommendation accuracy in terms of Recall(diff), for grp-pop(+-) and CKRM+SVD(+-) across different student sub-groups. Figure 5a shows the recommendation accuracy among different GPA-based student types, A vs B vs C. We notice that, first, CKRM+SVD(+-) outperforms grp-pop(+-) for all student groups. Second, we found that CKRM+SVD(+-) achieves the highest Recall(diff) for the type-B students, followed by type-A, and then by type-C. This could be due to the following reasons. After analyzing the training data, we found that the type-A and type-B students constitute $\sim$96% of the student population. After analyzing the average pairwise percentage of common courses and degree similarity among each GPA-based groups of students, as well as among pairs of different GPA-based groups, we found that type-C students follow more diverse sequencing for their degree plans that type-A or type-B students, as illustrated in Table 6, while there was no difference among the different groups in the average pairwise percentage of common courses. As discussed in Section 5.6.1., there is a high correlation between the pairwise degree similarity and recommendation accuracy. Since there is no



(a) Recommendation accuracy per student type.

(b) Recommendation accuracy per cohort.

Figure 5: Recommendation accuracy on different student sub-groups.

enough training data for the type-C students to learn their sequencing of the courses, this can explain why the recommendation accuracy for them was the lowest.

Figure 5b shows the recommendation accuracy among different student sub-groups based on their academic level. At the University of Minnesota, there are four academic levels, based on the number of both earned and transferred credits by the beginning of the semester: (1) freshman (≤ 30 credits), (2) sophomore (> 30 and ≤ 60 credits), (3) junior (> 60 and ≤ 90 credits), and senior (> 90 credits). First, we can notice that CKRM+SVD(+-) outperforms grp-pop(+-) across all student groups. Second we see that, as the student's academic level increases, and hence he/she has spent more years at the university and took more courses, both methods tend to achieve more accurate recommendations. This can be due to the following reasons. First, since we filter out the courses that have been previously taken by the student before making recommendations (see Section 4.3.), this means that as the student's academic level increases, there is a smaller number of candidate courses from which the recommendations are to be made. Second, for CKRM+SVD(+-), as the student takes more courses, his/her implicit profile that is computed by aggregating the embeddings of the previously-taken courses becomes more accurate.

6. Characteristics of Recommended Courses

An important question to any recommendation method is what the characteristics of the recommendations are. In this section, we study two important characteristics for the recommended courses, (i) the difficulty of courses (Section 6.1.), and (ii) their popularity (Section 6.2.) (RQ7).

Table 6: Average pairwise degree similarity between different pairs of GPA-based student groups.

Student Pair	Degree Similarity
A-B	0.597
A-C	0.535
B-C	0.534

Table 7: Statistics for the grades of all and recommended courses.

Course Set	Mean	Median	Std. Dev.
All	3.50	3.61	0.51
SVD(++)	3.24	3.24	0.27
SVD(+)	3.40	3.40	0.24
SVD(+-)	3.56	3.55	0.20

6.1. Course Difficulty

As our proposed grade-aware recommendation methods are trained to recommend courses that help students maintain or improve their GPA, these methods can be prone to recommending easier courses in which students usually achieve high grades. Here, we investigate whether this happens in our recommendations or not. Table 7 shows the grade statistics of all courses, as well as the courses recommended by all variations of grade-unaware and grade-aware SVD variations. The mean grade is 3.5 for all courses, while for the recommended courses, it is 3.24, 3.4, and 3.56, for SVD(++), SVD(+) and SVD(+-), respectively. These statistics show that the grade-aware SVD approaches tend to only slightly favor easier courses in their recommendations than the grade-unaware SVD approach.

6.2. Course Popularity

Since the university administrators need to make sure that students are enrolled in courses with different popularity, as there is a capacity for each course and classroom, course popularity is an important factor for course recommendations.

We also analyze the results of our models in terms of the popularity of the courses they recommend. Figure 6 shows the frequency of the actual good courses in the test set, as well as the frequency of the good courses recommended by both grp-pop(+-) and CKRM+SVD(+-)[6].

The figure shows that both grp-pop(+-) and CKRM+SVD(+-) recommend courses with different popularity[7], similar to the actual good courses taken by students. Comparing CKRM+-SVD(+-) to grp-pop(+-), we can notice that, grp-pop(+-) tends to recommend a higher number of the more popular courses, while CKRM+SVD(+-) recommends more of the less popular ones, which can be considered a major benefit for the latter method.

7. Discussion and Conclusion

In this paper, we proposed grade-aware course recommendation approaches for solving the course recommendation problem. The proposed approach aims to recommend to students good courses on which the student's expected grades will maintain or improve their overall GPA. We proposed two different approaches for solving the grade-aware course recommendation problem. The first approach ranks the courses by using an objective function that differentiates between sequences of courses that are expected to increase or decrease a student's GPA. The second approach combines the grades predicted by grade prediction methods in order to improve the

[6] Because we recommend $n_{(s,t)}$ courses, which is the total number of (good and bad) courses taken by student s in term t (see Section 4.3.), the number of recommendations can be higher than the number of actual good courses.

[7] Since we use a filtering technique before making recommendations, grp-pop(+-) can recommend courses with little popularity (see Section 4.3.)

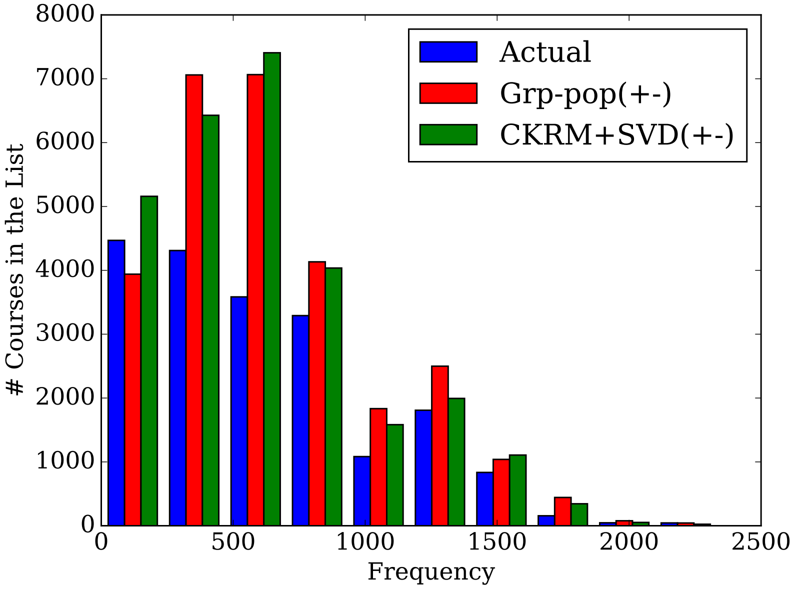


Figure 6: Popularity of the actual good courses, as well as courses recommended by grp-pop(+-) and CKRM+SVD(+-).

rankings produced by course recommendation methods. To obtain course rankings in the first approach, we adapted two widely-known representation learning techniques; one that uses the linear singular value decomposition model, while the other uses log-linear neural network-based models.

We conducted an extensive set of experiments on a large dataset obtained from 23 different majors at the University of Minnesota. The results showed that (i) the proposed grade-aware course recommendation approaches outperform grade-unaware recommendation methods in recommending more courses that increase the students' GPA and fewer courses that decrease it; (ii) the proposed representation learning-based approaches outperform competing approaches for grade-aware course recommendation; and (iii) the approaches that utilize both the good and bad courses and differentiate between them achieve comparable performance to combining grade prediction with the approaches that either utilize the good courses only or those that differentiate between good and bad courses.

We also provided an in-depth analysis of the recommendation accuracy across different majors and student groups. We found that our proposed approaches consistently outperformed the best baseline method across these majors and groups. We also analyzed the characteristics of the recommendations in terms of course difficulty and popularity. We found that our proposed grade-aware course recommendation approaches are not prone to recommending easy courses and that they recommend courses with high and low popularity in a similar manner. This shows the effectiveness of our proposed grade-aware approaches for course recommendation.

Time-to-degree is another important factor for academic success, which is the number of years or terms that the student enrolls in to finish his/her degree. An interesting research direction would be to investigate the effect of our recommendations on the time-to-degree, and accordingly, develop recommendation approaches that consider both the student's GPA and time-to-degree.

Acknowledgment

We would like to thank the anonymous reviewers for their valuable feedback on the original manuscript. This work was supported in part by NSF (1447788, 1704074, 1757916, 1834251), Army Research Office (W911NF1810344), Intel Corp, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute, http://www.msi.umn.edu.

References

Backenköhler, M., Scherzinger, F., Singla, A., and Wolf, V. 2018. Data-driven approach towards a personalized curriculum. In *Proceedings of the 11th International Conference on Educational Data Mining*. 246–251.

Bell, R., Koren, Y., and Volinsky, C. 2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '07. ACM, New York, NY, USA, 95–104.

Bendakir, N. and Aïmeur, E. 2006. Using association rules for course recommendation. In *Proceedings of the AAAI Workshop on Educational Data Mining*. Vol. 3. 1–10.

Bhumichitr, K., Channarukul, S., Saejiem, N., Jiamthapthaksin, R., and Nongpong, K. 2017. Recommender systems for university elective course recommendation. In *14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE, 1–5.

Braxton, J. M., Hirschy, A. S., and McClendon, S. A. 2011. *Understanding and Reducing College Student Departure: ASHE-ERIC Higher Education Report, Volume 30, Number 3*. Vol. 16. John Wiley & Sons.

Chen, S., Moore, J. L., Turnbull, D., and Joachims, T. 2012. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 714–722.

Cucuringu, M., Marshak, C. Z., Montag, D., and Rombach, P. 2017. Rank aggregation for course sequence discovery. In *International Workshop on Complex Networks and their Applications*. Springer, 139–150.

Elbadrawy, A. and Karypis, G. 2016. Domain-aware grade prediction and top-n course recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 183–190.

Elbadrawy, A., Studham, R. S., and Karypis, G. 2015. Collaborative multi-regression models for predicting students' performance in course activities. In *Proceedings of the 5th International Learning Analytics and Knowledge Conference*. 103–107.

Golub, G. H. and Reinsch, C. 1970. Singular value decomposition and least squares solutions. *Numerische mathematik 14,* 5, 403–420.

González-Brenes, J. P. and Mostow, J. 2012. Dynamic cognitive tracing: Towards unified discovery of student and cognitive models. In *Proceedings of the 5th International Conference on Educational Data Mining*. 49–56.

Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., and Sharp, D. 2015. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of* *the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1809–1818.

Grover, A. and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

Hagemann, N., O'Mahony, M. P., and Smyth, B. 2018. Module advisor: Guiding students with recommendations. In *Intelligent Tutoring Systems*, R. Nkambou, R. Azevedo, and J. Vassileva, Eds. Springer International Publishing, Cham, 319–325.

Hershkovitz, A., Gowda, S. M., and Corbett, A. T. 2013. Predicting future learning better using quantitative analysis of moment-by-moment learning. In *Proceedings of the 6th International Conference on Educational Data Mining*. 74–81.

Hu, Q. and Rangwala, H. 2018. Course-specific Markovian models for grade prediction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 29–41.

Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 873–882.

Hwang, C.-S. and Su, Y.-C. 2015. Unified clustering locality preserving matrix factorization for student performance prediction. *IAENG International Journal of Computer Science 42,* 3, 245–253.

Kena, G., Hussar, W., McFarland, J., de Brey, C., Musu-Gillette, L., Wang, X., Zhang, J., Rathbun, A., Wilkinson-Flicker, S., Diliberti, M., Barmer, A., Bullock Mann, F., and Dunlop Velez, E. 2016. The condition of education 2016. Tech. Rep. NCES 2016-144, National Center for Education Statistics.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 426–434.

Lan, A. S., Waters, A. E., Studer, C., and Baraniuk, R. G. 2014. Sparse factor analysis for learning and content analytics. *The Journal of Machine Learning Research 15,* 1, 1959–2008.

Le, Q. and Mikolov, T. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 1188–1196.

Lee, Y. and Cho, J. 2011. An intelligent course recommendation system. *SmartCR 1,* 1, 69–84.

Meier, Y., Xu, J., Atan, O., and Schaar, M. v. d. 2015. Personalized grade prediction: A data mining approach. In *IEEE International Conference on Data Mining*. 907–912.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

Morsy, S. and Karypis, G. 2017. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 552–560.

Morsy, S. and Karypis, G. 2019. A study on curriculum planning and its relationship with graduation gpa and time to degree. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*. ACM, 26–35.

PARAMESWARAN, A., VENETIS, P., AND GARCIA-MOLINA, H. 2011. Recommendation systems with complex constraints: A course recommendation perspective. *ACM Transactions on Information Systems (TOIS) 29,* 4, 20:1–20:33.

PARAMESWARAN, A. G. AND GARCIA-MOLINA, H. 2009. Recommendations with prerequisites. In *Proceedings of the Third ACM Conference on Recommender Systems*. ACM, 353–356.

PARAMESWARAN, A. G., GARCIA-MOLINA, H., AND ULLMAN, J. D. 2010. Evaluating, combining and generalizing recommendations with prerequisites. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 919–928.

PARAMESWARAN, A. G., KOUTRIKA, G., BERCOVITZ, B., AND GARCIA-MOLINA, H. 2010. Recsplorer: recommendation algorithms based on precedence mining. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. ACM, 87–98.

PARDOS, Z. A., FAN, Z., AND JIANG, W. 2019. Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User Modeling and User-Adapted Interaction*, 1–39.

PATEREK, A. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD Cup and Workshop*. Vol. 2007. 5–8.

PENNINGTON, J., SOCHER, R., AND MANNING, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

PEROZZI, B., AL-RFOU, R., AND SKIENA, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 701–710.

POLYZOU, A. AND KARYPIS, G. 2016. Grade prediction with course and student specific models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 89–101.

REDDY, S., LABUTOV, I., AND JOACHIMS, T. 2016. Latent skill embedding for personalized lesson sequence recommendation. *arXiv preprint*.

ROMERO, C., VENTURA, S., ESPEJO, P. G., AND HERVÁS, C. 2008. Data mining algorithms to classify students. In *Proceedings of the 1st International Conference on Educational Data Mining*. 8–17.

SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. 2000. Application of dimensionality reduction in recommender system a case study. In *Proceeding of WebKDD-2000 Workshop*.

SWEENEY, M., LESTER, J., RANGWALA, H., AND JOHRI, A. 2016. Next-term student performance prediction: A recommender systems approach. *Journal of Educational Data Mining 8,* 1, 22–51.

TANG, J., QU, M., WANG, M., ZHANG, M., YAN, J., AND MEI, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1067–1077.

THAI-NGHE, N., DRUMOND, L., HORVÁTH, T., AND SCHMIDT-THIEME, L. 2012. Using factorization machines for student modeling. In *UMAP Workshops*.

THAI-NGHE, N., HORVÁTH, T., AND SCHMIDT-THIEME, L. 2011. Factorization models for forecasting student performance. In *Proceedings of the 4th International Conference on Educational Data Mining*. 11–20.

WANG, P., GUO, J., LAN, Y., XU, J., WAN, S., AND CHENG, X. 2015. Learning hierarchical representation model for nextbasket recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 403–412.

XU, J., XING, T., AND VAN DER SCHAAR, M. 2016. Personalized course sequence recommendations. *IEEE Transactions on Signal Processing 64,* 20, 5340–5352.