Context-aware Nonlinear and Neural Attentive Knowledge-based Models for Grade Prediction

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

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

Grade prediction can help students and their advisers select courses and design personalized degree programs based on predicted future course performance. One of the successful approaches for accurately predicting a student's grades in future courses is Cumulative Knowledge-based Regression Models (CKRM). CKRM learns shallow linear models that predict a student's grades as the similarity between his/her knowledge state and the target course. However, there can be more complex interactions among prior courses taken by a student, which cannot be captured by the current linear CKRM model. Moreover, CKRM and other grade prediction methods ignore the effect of concurrently-taken courses on a student's performance in a target course. In this paper, we propose context-aware nonlinear and neural attentive models that can potentially better estimate a student's knowledge state from his/her prior course information, as well as model the interactions between a target course and concurrent courses. Compared to the competing methods, our experiments on a large real-world dataset consisting of more than 1.5 million grades show the effectiveness of the proposed models in accurately predicting students' grades. Moreover, the attention weights learned by the neural attentive model can be helpful in better designing their degree plans.

Keywords: grade prediction, degree plans, knowledge-based models, nonlinear models, neural attentive models, undergraduate education

1. Introduction

The average six-year graduation rate across four-year higher-education institutions in the U.S. has been around 59% over the past 15 years (Kena et al., 2016; Braxton et al., 2011), while less than half of U.S. 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.

One approach for improving graduation and retention rates is to help students make good selections about the courses they register for in each term, such that the knowledge they have acquired in the past would prepare them to succeed in the next-term enrolled courses. Polyzou and Karypis (2016) proposed course- and student-specific linear models that learns the importance (or weight) or each previously-taken term towards accurately predicting the grade in a future course. One limitation of this approach is that in order to make accurate predictions, the model needs to have sufficient training data for each (prior, target) tuple. Morsy and Karypis (2017) developed Cumulative Knowledge-based Regression Models (CKRM) that also build on the idea of accumulating knowledge over time. CKRM predicts the student's grades as the similarity between his/her knowledge state and the target course. Both the student's knowledge state and the target course are represented as low-dimensional embedding vectors and the similarity between them is modeled by their inner product. The student's knowledge state is implicitly computed as a linear combination of the so-called provided knowledge component vectors of the previously-taken courses, weighted by his/her grades in them. Though CKRM was shown to provide state-of-the-art grade prediction accuracy, it is limited in that it learns shallow linear models that may not be able to model the complex interactions among prior courses, or to model the different contribution/importance of each prior course towards each target course. In addition, it does not consider the effect that concurrently-taken courses can have on a student's performance in a target course.

In this work, we develop context-aware nonlinear and neural attentive models that improve upon CKRM from two perspectives. First, they can model the nonlinear interactions among prior courses as well as the different contributions of priors course to estimate the student's knowledge state more accurately, by using two different approaches. In the first approach, we hypothesize that each course provides a set of knowledge components at a specific knowledge level. It uses a nonlinear model that aggregates the weighted prior course embeddings by employing a maximum-based pooling layer along each component of the prior courses' embeddings. In the second approach, we hypothesize that prior courses contribute differently towards a target course, and that some of them may not be relevant to it. Motivated by the success of neural attentive networks in different fields (He et al., 2018; Mei et al., 2018; He and Chua, 2017; Bahdanau et al., 2014; Parikh et al., 2016; Xiao et al., 2017), we learn attention weights for the prior courses that denote their importance to a target course using two different activation functions. Second, the proposed models consider the effect of the concurrently-taken courses while predicting a student's grade in a target course. We hypothesize that the knowledge provided by concurrent courses affect the knowledge required by a target course. We model the interaction between the concurrent and target course, in terms of the synergy and competition among them, using nonlinear and neural attentive models, as well.

The main contributions of this work are as follows:

1. We propose context-aware nonlinear and neural attentive knowledge-based models for grade prediction that improve upon the linear CKRM model by: (i) using nonlinear and neural attentive models to capture the different contribution of each prior course while aggregating their embeddings to compute a student's knowledge state, as well as their different contributions towards each target course; and (ii) modeling the effect of the concurrently-taken courses using nonlinear and neural attentive models. To our knowledge, this is the first work to model the effect of the concurrently-taken courses in grade prediction.

2. We leverage the recent sparsemax activation function for the attention mechanism in the neural attentive models that produces sparse attention weights instead of soft attention weights.

3. We performed an extensive experimental evaluation on a real world dataset obtained from a large public university that spans a period of 16 years and consists of ∼1.5 million grades. The results show that: (i) the proposed context-aware nonlinear and neural attentive models outperform other baseline methods, including the previously-developed CKRM method, with statistically significant improvements; (ii) the context-aware nonlinear model outperforms the context-aware neural attentive model and all baselines in making less severe under-predictions; (iii) estimating a student's knowledge state via a nonlinear or neural attentive model significantly outperforms estimating it via a linear model; (iv) learning sparse attention weights for the neural attentive model outperforms learning soft weights; (v) modeling the interactions between a target course and concurrent courses significantly improves the performance of the nonlinear model and gives similar performance for the neural attentive model; and (vi) the neural attentive model was able to uncover the listed and hidden pre-requisite courses for target courses.

2. Definitions and Notations

Boldface uppercase letters will be used to represent matrices (e.g., $\mathbf{G}$) and boldface lowercase letters to represent row vectors, (e.g., $\mathbf{p}$). The ith row of matrix $\mathbf{P}$ is represented as $\mathbf{p}_i^T$, and its jth column is represented as $\mathbf{p}_j$. The entry in the ith row and jth column of matrix $\mathbf{G}$ is denoted as $g_{i,j}$. A predicted value is denoted by having a hat over it (e.g., $\hat{g}$).

Matrix $\mathbf{G}$ will represent the $m \times n$ student-course grades matrix, where $g_{s,c}$ denotes the grade that student s obtained in course c, relative to his/her average previous grade. Following the row-centering technique that was first proposed by Polyzou and Karypis (2016), we subtract each student's grade from his/her average previous grade, since this was shown to significantly improve the prediction accuracy of different models. As there can be some students who achieved the same grades in all their prior courses, and hence their relative grades will be zero, in this case, we assigned a small value instead, i.e., 0.01. This is to prevent a prior course from not being considered in the model computation. A student s enrolls in sets of courses in consecutive terms, numbered relative to s from 1 to the number of terms in he/she has enrolled in the dataset. A set $\mathcal{T}_{s,w}$ will denote the set of courses taken by student s in term w.

3. Related Work

In this section, we review and identify several research areas that are highly relevant to our work.

3.1. Grade Prediction Methods

Grade prediction approaches for courses not yet taken by students have been extensively explored in the literature (Elbadrawy and Karypis, 2016; Hu and Rangwala, 2018; Morsy and Karypis, 2017; Polyzou and Karypis, 2016; Ren et al., 2017; Ren et al., 2018; Sweeney et al., 2016). Sweeney et al. (2016) investigated the use of recommender systems techniques for grade prediction. They employed different methods, such as matrix factorization, random forests and linear regression. Elbadrawy and Karypis (2016) developed several grade prediction and course recommendation methods that make use of the student- and course-based academic grouping information. Students can be grouped based on the colleges they attend, their declared majors and/or their academic levels. Courses can be grouped based on their subjects and/or levels, e.g., CSCI 5481 belongs to the Computer Science subject and level 5. The authors hypothesized that grouping students and/or courses into one of these finer-grained groups and incorporating this information into matrix factorization, user-based collaborative filtering, and popularity-based ranking, give more accurate grade prediction and recommendation. To this end, the authors introduced the use of local student and course biases into the aforementioned methods for both grade prediction and course recommendation. Using the finer-grained grouping improved the recommendation accuracy, but did not add much to the grade prediction accuracy.

3.1.1. Course-Specific Regression Models (CSR)

A more recent and natural way to model the grade prediction problem is to model the way the academic degree programs are structured. Each degree program would require students to take courses in a specific sequencing such that the knowledge acquired in previous courses are required for a student to perform well in future courses. Polyzou and Karypis (2016) developed course-specific linear regression models (CSR) that build on this idea. A student's grade in a course is estimated as a linear combination of his/her grades in previously-taken courses, with different weights learned for each (prior, target) course pair. For a student s and a target course j, the predicted grade is estimated as:

$$\hat{g}_{s,j} = b_j + \sum_{i \in \mathcal{P}} w_{i,j}\, g_{s,i}, \tag{1}$$

where b_j is the bias term for course j, $w_{i,j}$ is the weight of course i towards predicting the grade of course j, $g_{s,i}$ is the grade of student s in course i, and $\mathcal{P}$ is the set of courses taken by s prior to taking course j. To achieve high prediction accuracy, CSR requires sufficient training data for each (prior, target) pair, which can hinder these models from good generalization.

3.1.2. Cumulative Knowledge-based Regression Models (CKRM)

Morsy and Karypis (2017) developed Cumulative Knowledge-based Regression Models (CKRM), which also models a student's performance in a future course based on his/her performance in the previously-taken courses. It assumes that there is a space of knowledge components that corresponds to the fundamental concepts associated with each discipline. This space can be explicitly defined, for instance, in the case of Computer Science, it can correspond to the union of topics ACM's Computer Science curriculum[1]), or it can be latent and identified during training, which is the approach used in this paper.

CKRM assumes that a course *provides* a subset of these knowledge components to the student that takes that course and that a course *requires* the knowledge of some of these components from the student in order to perform well in it. CKRM models the course-provided and course-required knowledge components by associating a pair of vector $\mathbf{p}_i$ and $\mathbf{r}_i$ to each course, respectively.

A student by taking a course acquires its knowledge components in a way that depends on his/her grade in that course. The overall knowledge acquired by a student after taking a set of courses is then represented by a knowledge state vector that is computed as the sum of the knowledge component vectors of those courses, weighted by his/her grades in them. The

[1] https://www.acm.org/education/curricula-recommendations

knowledge state vector for student s at term t can be expressed as follows:

$$\mathbf{k}_{s,t} = \sum_{w=1}^{t-1} \xi(s,w,t) \sum_{i \in \mathcal{T}_{s,w}} \Big(g_{s,i}\, \mathbf{p}_i \Big), \tag{2}$$

where $g_{s,i}$ is the grade that student s obtained on course i, and $\xi(s,w,t)$ is a time-based exponential decaying function designed to de-emphasize courses that were taken a long time ago.

Given a student's knowledge state vector prior to taking a course and that course's required knowledge component vector, $\mathbf{r}_j$, CKRM estimates that student's expected grade in that course as the inner product of these two vectors, i.e.,

$$\hat{g}_{s,j} = b_j + \mathbf{k}_{s,t}^T\, \mathbf{r}_j, \tag{3}$$

where b_j is as defined in Eq. 1, and $\mathbf{k}_{s,t}$ is the corresponding knowledge state vector.

When the knowledge component space is latent, the course-provided and course-required vectors as well as the course biases are estimated directly from the data using an alternating optimization approach.

3.2. Neural Attentive Models

Neural networks have been used extensively in many fields, including, but not limited to: natural language processing (Bahdanau et al., 2014; Parikh et al., 2016) and recommender systems (He and Chua, 2017; He et al., 2018; Mei et al., 2018; Xiao et al., 2017). The attention mechanism has been recently introduced to neural network modeling and was shown to improve the performance of different models. Instead of aggregating the input object embeddings via a summation or mean pooling function, which assumes equal contribution of all objects, the idea is to allow the selected objects to contribute differently when compressing them to a single representation. Neural attentive networks have been successfully applied in many recommendation system techniques, such as factorization machines (He and Chua, 2017; Xiao et al., 2017), item-based collaborative filtering (He et al., 2018), and user-based collaborative filtering (Chen et al., 2017).

Part of our work relies on the attention mechanism, and leverages several advances in this area. The most commonly-used activation function for the attention mechanism is the softmax function, which is easily differentiable and gives soft posterior probabilities that normalize to 1. A major disadvantage of the softmax function is that it assumes that each object contributes to the compressed representation, which may not always hold in some domains. To solve this, we need to output sparse posterior probabilities and assign zero to the irrelevant objects. Martins and Astudillo (2016) proposed the sparsemax activation function, which has the benefit of assigning zero probabilities to some output variables that may not be relevant for making a decision. This is done by defining a threshold, below which small probability values are truncated to zero. We also leverage the controllable sparsemax activation function recently proposed by Laha et al. (2018) that controls the desired degree of sparsity in the output probabilities. This is done by adding an L2 regularization term that is to be maximized in the loss function. This will potentially encourage larger probability values for some objects, moving the rest to zero.

4. Nonlinear and Neural Attentive Knowledge-based Models

CKRM (Morsy and Karypis, 2017) uses shallow linear models to aggregate the prior courses' embeddings taken by a student in order to estimate his/her knowledge state. We propose two alternative approaches to emphasize the courses that may be more related to the target course: a nonlinear maximum knowledge-based model (Section 4.1.), and a neural attentive knowledge-based model (Section 4.2.).

4.1. Maximum Knowledge-based Models

In this section, we develop a **MA**ximum **K**nowledge-based model (MAK), which estimates a student's knowledge state by applying a maximum-based pooling layer on the prior courses. We use CKRM as the underlying model (see Section 3.1.2.).

4.1.1. Motivation

Undergraduate degree programs are structured in a way such that earlier courses provide basic knowledge that is built upon in the later courses that provide more advanced knowledge. Consider two courses offered by a Computer Science department: Introduction to Programming in C/C++ and Advanced Programming Principles. We would expect that the introduction to programming course provides basic knowledge to programming to freshman students who may be exposed to programming for the first time. The advanced programming course builds on the knowledge acquired by the introductory course, and provides more advanced knowledge components related to programming principles and programming languages. When a student takes the introductory and then the advanced course, he/she can only acquire the maximum knowledge components provided by both of them, since each course provides very similar knowledge components, but at a different knowledge level.

4.1.2. Maximum-based Pooling Layer for Prior Courses

Based on our hypothesis explained above, we can estimate a student s's knowledge state at the beginning of term t as follows:

$$\mathbf{k}_{s,t} = \begin{bmatrix} \max\limits_i \Big(\xi(s, w_{s,i}, t)\ g_{s,i}\ p_{i,1} \Big) \\ . \\ . \\ . \\ \max\limits_i \Big(\xi(s, w_{s,i}, t)\ g_{s,i}\ p_{i,d} \Big) \end{bmatrix}, \forall i \in \mathcal{T}_{s,y} \text{ for } y = 1, \ldots, t-1, \tag{4}$$

where $w_{s,i}$ is the relative term number when s took course i, $\xi(s, w_{s,i}, t)$ is a time-based exponential decaying function, $p_{i,z}$ is the zth entry in $\mathbf{p}_i$, $\mathcal{T}_{s,y}$ is the set of courses taken by s in term y, and d is the embedding size of the vector $\mathbf{p}$.

4.2. Neural Attentive Knowledge-based Models

In this section, we develop a **N**eural **A**ttentive **K**nowledge-based model (NAK), which applies an attention mechanism on prior courses to learn individual weights for them that represent their importance to a target course before aggregating them to estimate a student's knowledge state. We also use CKRM as the underlying model (see Section 3.1.2.).

Table 1: Sample of prior and target courses for a Computer Science student at the University of Minnesota.

Prior Courses	Target Course
Calculus I, Beginning German, Operating Systems, Intermediate German I, University Writing, Introductory Physics, Poetics in Film, Program Design & Development, Philosophy, Linear Algebra, Internet Programming, Stone Tools to Steam Engines, Advanced Programming Principles, Computer Networks	Intermediate German II
	Probability & Statistics
	Algorithms & Data Structures

4.2.1. Motivation

Consider a student who is declared in a Computer Science major and is in his/her second or third year in college. Table 1 shows the set of prior courses that this student has already taken and the set of courses that this student is planning on taking the next term. We can see from the courses' names that there are courses that are strongly related to each target course and other courses that are irrelevant to it. For instance, it is reasonable to expect that the Intermediate German II course is more related to the Intermediate German I course than any of the other courses that the student has already taken. Along the same lines, we expect that the Algorithms and Data Structures course is more related to other Computer Science courses, such as the Advanced Programming Principles and the Program Design and Development courses. Learning the different contribution/importance of each prior course towards a target course can make grade prediction more accurate.

4.2.2. Attention-based Pooling Layer for Prior Courses

In order to learn the different contributions of prior courses in estimating a student's grade in a future course, we can employ the CSR technique (see Section 3.1.1.) that learns the importance of each prior course in estimating the grade of each future course. Thus, we would estimate a knowledge state vector for each target course j, using the following equation:

$$\mathbf{k}_{s,t,j} = \sum_{w=1}^{t-1} \sum_{i \in \mathcal{T}_{s,w}} \left(a^{p}_{i,j}\, g_{s,i}\, \mathbf{p}_i \right), \tag{5}$$

where $a^{p}_{i,j}$ is a learnable parameter that denotes the attention weight of course i in contributing to student s's knowledge state when predicting his/her grade in course j. However, this solution requires sufficient training data for each (i, j) pair in order to be considered an accurate estimation.

In order to be able to have accurate attention weights between all pairs of prior and target courses, even the ones that do not appear together in the training data, we propose to use the attention mechanism that was recently used in neural networks (Bahdanau et al., 2014; Vaswani et al., 2017). The main idea is to estimate the attention weight $a^{p}_{i,j}$ from the embedding vectors for courses i and j.

In order to compute the similarity between the embeddings of prior course i and target course j, we use a single-layer perceptron as follows:

$$z^{p}_{i,j} = \mathbf{h}^{pT}\mathrm{RELU}(\mathbf{W}^{p}(\mathbf{q}_i \odot \mathbf{r}_j) + \mathbf{b}^{p}), \tag{6}$$

where $\mathbf{q}_i = g_{s,i}\mathbf{p}_i$ denotes the embedding of the prior course i, weighted by s's grade in it, $\mathbf{W}^p \in \mathcal{R}^{l \times d}$ and $\mathbf{b}^p \in \mathcal{R}^l$ denote the weight matrix and bias vector that project the input into a hidden layer, respectively, $\odot$ denotes the Hadamard product, and $\mathbf{h}^p \in \mathcal{R}^l$ is a vector that projects the hidden layer into an output attention weight, where d and l denote the number of dimensions of the embedding vectors and attention network, respectively. RELU denotes the Rectified Linear Unit activation function that is usually used in neural attentive networks.

After computing the affinity vector $\mathbf{z}^p$ that represents the similarity between each prior course and the target course, an activation function is used to convert $\mathbf{z}^p$ to attention weights that follow a probability distribution that sum up to 1. In the remainder of this section, we explain the two activation functions that we used: the softmax and sparsemax activation functions.

SOFTMAX ACTIVATION FUNCTION The most common activation function used for computing these attention weights is the softmax function (Vaswani et al., 2017). Given a vector of real weights $\mathbf{z}$, the softmax activation function converts it to a probability distribution, which is computed component-wise as follows:

$$\text{softmax}_i(\mathbf{z}) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}. \tag{7}$$

We will refer to this method as **NAK(soft)**.

SPARSEMAX ACTIVATION FUNCTION Although the softmax activation function has been used to design attention mechanisms in many domains (Bahdanau et al., 2014; He and Chua, 2017; He et al., 2018; Mei et al., 2018; Parikh et al., 2016; Xiao et al., 2017), we believe that using it for grade prediction can degrade the accuracy of prediction. Since a student enrolls in several courses, and each course requires knowledge from one or a few other courses, we hypothesize that some of the prior courses should have no effect, i.e., zero attention, towards predicting a target course's grade. We thus leverage a recent advance, the sparsemax activation function (Martins and Astudillo, 2016), to learn sparse attention weights. The idea is to define a threshold, below which small probability values are truncated to zero. Let $\triangle^{K-1} := \{\mathbf{x} \in \mathbb{R}^K | \mathbf{1}^T\mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}$ be the $(K-1)$-dimensional simplex. The sparsemax activation function tries to solve the following equation:

$$\text{sparsemax}(\mathbf{z}) = \underset{\mathbf{x} \in \triangle^{K-1}}{\text{argmin}} \; \|\mathbf{x} - \mathbf{z}\|^2, \tag{8}$$

which, in other words, returns the Euclidean projection of the input vector $\mathbf{z}$ onto the probability simplex. We will refer to this method as **NAK(sparse)**.

In order to obtain different degrees of sparsity in the attention weights, Laha et al. (2018) developed a generic probability mapping function for the sparsemax activation function, which they called **sparsegen**, and is computed as follows:

$$\text{sparsegen}(\mathbf{z};\gamma) = \text{argmin} \; \|\mathbf{x} - \mathbf{z}\|^2 - \gamma\|\mathbf{x}\|^2, \tag{9}$$

where $\gamma < 1$ controls the L2 regularization strength of $\mathbf{x}$. An equivalent formulation for sparsegen was formed as:

$$\text{sparsegen}(\mathbf{z};\gamma) = \text{sparsemax}\big(\frac{\mathbf{z}}{1-\gamma}\big), \tag{10}$$

which, in other words, applies a temperature parameter to the original sparsemax function. Varying this temperature parameter can change the degree of sparsity in the output variables. By setting $\gamma = 0$, sparsegen becomes equivalent to sparsemax.

5. Context-aware Nonlinear and Neural Attentive Knowledge-based Models

Another limitation of existing grade prediction methods is that they ignore the effect of concurrently-taken courses. We hypothesize that the concurrent courses can affect a student's grade in a target course. For instance, the knowledge provided by concurrent courses can help a student in better understanding the material given in a target course. In addition, since a student's time is limited, the effort that he/she spends on a target course is affected by the difficulty of courses taken concurrently with it. These interactions create synergy and/or competition among a target course and concurrently-taken courses We thus estimate a context-aware embedding for a target course that we would like to predict a student's grade in, given the courses taken concurrently with it. We utilize the proposed MAK and NAK models (Section 4.) as our underlying models.

To model the interactions between a target course and other courses taken concurrently with it, we estimate a context-aware embedding for that target course as follows:

$$\mathbf{e}_{j,w} = \mathbf{x}_{j,w} \odot \mathbf{r}_j, \tag{11}$$

where $\mathbf{x}_{j,w}$ denotes the aggregated embedding of the courses that are taken concurrently with j in term w, $\odot$ denotes the Hadamard product, and $\mathbf{r}_j$ denotes the required knowledge component vector for target course j. To aggregate the concurrent courses' embeddings, we use nonlinear and neural attentive models similar to the ones developed in Sections 4.1. and 4.2., respectively.

5.1. Context-aware Maximum Knowledge-based Models

In this section, we develop a **C**ontext-aware **MA**ximum **K**nowledge-based model (CMAK), which models the interactions between a target and concurrent courses using MAK (Section 4.2.) as the underlying model.

The aggregated embedding of the courses that are taken concurrently with j in term w is estimated by applying a maximum-based pooling layer on them, similar to how we aggregated the prior courses' embeddings for the MAK model (Section 4.1.), and is computed as:

$$\mathbf{x}_{j,t} = \begin{bmatrix} \max\limits_i p_{i,1} \\ . \\ . \\ . \\ \max\limits_i p_{i,d} \end{bmatrix}, \forall i \in \mathcal{T}_{\{s,t\} \setminus \{j\}}, \tag{12}$$

where: $\mathbf{p}_{i,l}$ denotes the lth entry in the $\mathbf{p}_i$ vector, where $\mathbf{p}_i$ denotes the embedding for concurrent course i. Note that we use the same embedding vector $\mathbf{p}_i$ for representing both a prior and a concurrent course.

5.2. Context-aware Neural Attentive Knowledge-based Models

In this section, we develop a **C**ontext-aware **N**eural **A**ttentive **K**nowledge-based model (CNAK), which models the interactions between a target and concurrent courses using NAK (Section 4.2.) as the underlying model.

To aggregate the concurrent courses' embeddings, we employ an attention mechanism on them to learn the different contributions of each of them towards the target course, similar to how we aggregated the prior courses' embeddings for the NAK model (Section 4.2.2.). The aggregated embedding of the courses that are taken concurrently with j in term w is computed as:

$$\mathbf{x}_{j,w} = \sum_{i \in \mathcal{T}_{\{s,w\}\setminus\{j\}}} a^x_{i,j} \mathbf{p}_i, \tag{13}$$

where $a^x_{j,t}$ is the attention weight for the concurrent course j, and can be computed using the softmax (Eq. 7) or sparsegen (Eq. 10) activation function. The affinity between concurrent course i and target course j is computed in a similar way as in Eq. 6, i.e.,

$$z^x_{i,j} = \mathbf{h}^{xT}\text{RELU}(\mathbf{W}^x(\mathbf{p}_i \odot \mathbf{r}_j) + \mathbf{b}^x), \tag{14}$$

where $\mathbf{W}^x \in \mathcal{R}^{l \times d}$, $\mathbf{b}^x \in \mathcal{R}^l$ and $\mathbf{h}^x \in \mathcal{R}^l$ denote the attention network parameters for the concurrent courses, similar to the ones defined in Eq. 6, and $\odot$ denotes the Hadamard product.

6. Grade Prediction

Given a student s's representation at the beginning of term t and a target course j's representation that he/she is interested in taking, we can estimate s's grade in j for the different proposed methods in a similar way to CKRM as follows:

- Using MAK:

  $$\hat{g}_{s,j} = b_j + \mathbf{k}^T_{s,t}\,\mathbf{r}_j, \tag{15}$$

  where: $\mathbf{k}_{s,t}$ is as defined in Eq. 4 and b_j and $\mathbf{r}_j$ are as defined in Eq. 3.

- Using NAK:

  $$\hat{g}_{s,j} = b_j + \mathbf{k}^T_{s,t,j}\,\mathbf{r}_j, \tag{16}$$

  where: $\mathbf{k}_{s,t,j}$ is as defined in Eq. 5 and b_j and $\mathbf{r}_j$ are as defined in Eq. 15.

- Using CMAK:

  $$\hat{g}_{s,j} = b_c + \mathbf{k}^T_{s,t}\,\mathbf{e}_{j,t}, \tag{17}$$

  where: b_j is as defined in Eq. 15, $\mathbf{k}_{s,t}$ is as defined in Eq. 4, and $\mathbf{e}_{j,t}$ is as defined in Eq. 11.

- Using CNAK:

  $$\hat{g}_{s,j} = b_c + \mathbf{k}^T_{s,t,j}\,\mathbf{e}_{j,t}, \tag{18}$$

  where: b_j is as defined in Eq. 15, $\mathbf{k}_{s,t,j}$ is as defined in Eq. 5, and $\mathbf{e}_{j,t}$ is as defined in Eq. 11.

7. Model Optimization

We use the mean squared error (MSE) loss function to estimate the parameters of all our proposed models. We minimize the following regularized MSE loss:

$$L = -\frac{1}{2N} \sum_{s,c \in \mathbf{G}} (g_{s,c} - \hat{g}_{s,c})^2 + \lambda ||\Theta||^2, \tag{19}$$

where N is the number of grades in $\mathbf{G}$. The hyper-parameter λ controls the strength of L2 regularization to prevent overfitting, and $\Theta = \{\{\mathbf{b}\}, \{\mathbf{p}_i\}, \{\mathbf{r}_i\}\}$ denotes the learnable parameters for the MAK and CMAK models, $\Theta = \{\{\mathbf{b}\}, \{\mathbf{p}_i\}, \{\mathbf{r}_i\}$, $\mathbf{W}^p$, $\mathbf{b}^p$, $\mathbf{h}^p\}$, denotes the learnable parameters for the NAK model, and $\Theta = \{\{\mathbf{b}\}, \{\mathbf{p}_i\}, \{\mathbf{r}_i\}$, $\mathbf{W}^p$, $\mathbf{b}^p$, $\mathbf{h}^p$, $\mathbf{W}^x$, $\mathbf{b}^x$, $\mathbf{h}^x\}$ denotes the learnable parameters for the CNAK model, where $\mathbf{W}^p$, $\mathbf{b}^p$, and $\mathbf{h}^p$ denote the attention mechanism parameters for the prior courses, and $\mathbf{W}^x$, $\mathbf{b}^x$, and $\mathbf{h}^x$ denote the attention mechanism parameters for the concurrent courses.

The optimization problem is solved using AdaGrad algorithm (Duchi et al., 2011), which applies an adaptive learning rate for each parameter. It randomly draws mini-batches of a given size from the training data and updates the related model parameters. The source code for the proposed methods is available at: `https://github.com/KarypisLab/grade-prediction`.

8. Evaluation Methodology

8.1. Dataset

The data used in our experiments was obtained from the University of Minnesota (UMN), which includes 96 majors from 10 different colleges, and spans the years 2002 to 2017. At UMN, the letter grading system used is A–F, which is converted to the 4–0 scale using the standard letter grade to GPA conversion. We row-centered the student's grades in each term around his/her GPA achieved in previous terms, which was shown to significantly improve the prediction performance in Polyzou and Karypis (2016). We removed any grades that were taken as pass/fail. The final dataset includes $54,269$ students, $5,824$ courses, and $1,561,145$ grades in total.

8.2. Generating Training, Validation and Test Sets

At UMN, there are three terms, Fall, Summer and Spring. We used the data from 2002 to Spring 2015 (inclusive) as the training set, the data from Spring 2016 to Fall 2016 (inclusive) as the validation set, and the data from Summer 2016 to Summer 2017 (inclusive) as the test set. For a target course taken by a student to be predicted, that student must have taken at least four courses prior to the target course, in order to have sufficient data to compute the student's knowledge state vector. We excluded any courses that do not appear in the training set from the validation and test sets.

8.3. Baseline Methods

We compared the performance of the proposed methods against the following grade prediction methods:

1. **Matrix Factorization (MF):** This method predicts the grade for student s in course i as:

$$\hat{g}_{s,i} = \mu + sb_s + cb_i + \mathbf{u}_s^T \, \mathbf{v}_i, \tag{20}$$

   where μ, sb_s and cb_i are the global, student and course bias terms, respectively, and $\mathbf{u}_s$ and $\mathbf{v}_i$ are the student and course latent vectors, respectively. We used the squared loss function with L2 regularization to estimate this model.

2. **KRM(sum):** This is the CKRM method described in Section 3.1.2., and the underlying model for our proposed models.

3. **KRM(avg):** This is similar to the KRM(sum) method, except that the prior courses' embeddings are aggregated with mean pooling instead of summation. It was shown in later studies, e.g. Ren et al. (2018), that it performs better than KRM(sum).

We implemented KRM(sum) and KRM(avg) with a neural network architecture and optimization similar to that of the proposed methods.

8.4. Model Selection

We performed an extensive search on the parameters of the proposed and baseline models to find the set of parameters that gives us the best performance for each model.

For all proposed and competing models, the following parameters were used. The number of latent dimensions for course embeddings was chosen from the set of values: {8, 16, 32}. The L2 regularization parameter was chosen from the values: {1e-5, 1e-7, 1e-3}. Finally, the learning rate was chosen from the values: {0.0007, 0.001, 0.003, 0.005, 0.007}. For the proposed NAK and CNAK models, the number of latent dimensions for the MLP attention mechanism was selected in the range [1, 4]. For the sparsegen activation function in NAK and CNAK, the L2 regularization parameter γ was chosen from the values: {0.5, 0.9}. For KRM(sum), KRM(avg), MAK and CMAK, the time-decaying parameter λ was chosen from the set of values: {0, 0.3, 0.5, 0.7, 1.0}.

The training set was used for estimating the models, whereas the validation set was used to select the best performing parameters in terms of the overall MSE of the validation set.

8.5. Evaluation Methodology

The grading system used by UMN uses a 12 letter grade system (i.e., A, A-, B+, . . . F). We will refer to the difference between two successive letter grades (e.g., B+ vs B) as a *tick*. We converted the predicted grades into their closest letter grades. We assessed the performance of the different approaches based on the Root Mean Squared Error (RMSE) as well as how many ticks away the predicted grade is from the actual grade, which is referred to as *Percentage of Tick Accuracy*, or PTA. We computed the percentage of grades predicted with no error (zero tick), within one tick, and within two ticks, which will be referred to as PTA0, PTA1, and PTA2, respectively. In general, the grades that are predicted with at most one or two ticks of error are sufficiently accurate for the task of course selection.

In addition, we also report the percentage of grades predicted with severe errors. We report two metrics: (i) severe under-predictions; and (ii) severe over-predictions. *Severe under-predictions* will refer to the percentage of grades that are predicted with three or more tick errors lower than the actual corresponding grades. A severe under-prediction for a student in a target

Table 2: Comparison with baseline methods.

Model	RMSE (↓)	PTA0 (↑)	PTA1 (↑)	PTA2 (↑)
MF	0.724	25.7	58.6	79.5
KRM(sum)	0.584	32.6	70.1	87.7
KRM(avg)	0.584	34.9	70.6	87.7
CMAK	0.548† (6.2)	35.1 (0.6)	73.4 (4.0)	89.8 (2.4)
CNAK	0.569† (2.6)	35.5† (1.7)	72.0 (2.0)	88.7 (1.1)

Note. Underlined entries represent the best performance in each metric. † denotes statistical significance over the best baseline model, using the Student's t-test with a p-level < 0.05. Numbers in parentheses denote the percentage of improvement over the best baseline value in each metric.

course can result in an opportunity loss for that student who might falsely think that he/she is not well qualified for taking that course. *Severe over-predictions* will refer to the percentage of grades that are predicted with three or more tick errors higher than the actual corresponding grades. A severe over-prediction for a student in a course can motivate that student to take that course, incorrectly believing that he/she is well-prepared for taking it and will perform well in it. This might cause a decrease in the student's GPA or having to repeat that course at a later time.

9. Experimental Results

We present the results of our experiments to answer the following questions:

RQ1. How do the proposed context-aware nonlinear and neural attentive models compare against the competing methods?

RQ2. What is the impact of estimating a student's knowledge state via a nonlinear or neural attentive model?

RQ3. What is the impact of modeling the effect of concurrent courses on a student's performance in a target course?

RQ4. Are we able to derive any insights about the importance of different prior courses to target courses from the neural attentive, i.e., NAK, model?

9.1. Performance against Competing Methods

Table 2 shows the performance of the proposed models against the competing models (**RQ1**). Among the baseline methods, both KRM(sum) and KRM(avg) outperform MF. KRM(avg) outperforms KRM(sum) in predicting grades within no and one tick errors. Among all competing and proposed methods, the proposed CMAK and CNAK models outperform all baseline methods, with statistically significant improvements in some metrics, namely the RMSE and PTA0 metrics. CMAK and CNAK achieve 6.2% and 2.6% lower (better) RMSE, and 2.4% and 1.1%

Table 3: Severe under- and over-predictions by baseline and proposed models.

Model	Severe Under-predictions (↓)	Severe Over-predictions (↓)
KRM(sum)	5.4	6.9
KRM(avg)	5.6	6.7
CMAK	3.9 (27.4%)	6.3 (5.4%)
CNAK	4.9 (9.1%)	6.4 (3.4%)

Note. Underlined entries represent the best performance in each metric. Numbers in parentheses denote the percentage of improvement over the best baseline value in each metric.

more accurate predictions within two tick errors, respectively, than the best performing baseline method. This shows the effectiveness of the proposed context-aware nonlinear and neural attentive models in more accurately predicting the grades of students in their future courses than all competing methods. Comparing CMAK with CNAK, we see that CMAK outperforms CNAK, achieving 3.7% lower RMSE, and 1.2% more accurate predictions within two tick errors.

Table 3 shows the percentage of severe under- and over-predictions that were made by the different baseline and proposed models, denoting the grades that were predicted with three or more tick errors lower and higher than the actual grades, respectively. Severe under-predictions can result in an opportunity loss for students, urging them not to take these under-predicted courses in fear of lowering their GPAs. Severe over-predictions can result in urging them to take these over-predicted courses that they may not be well-prepared for and may lower their GPAs. For the severe under-predictions, both CMAK and CNAK outperform the KRM variants, achieving 27% and 9% less severe under-predictions. For the severe over-predictions, both CMAK and CNAK also outperform the KRM variants, achieving 5% and 3% less severe over-predictions. Comparing CMAK with CNAK, we see that CMAK outperforms CNAK, achieving 20% less severe under-predictions, and 2% less severe over-predictions. Since the grades in the data are row-centered around the students' average grades and a course bias term is learned for each course, it is hard for all these models to prevent severe over-predictions from occurring.

9.2. Effect of Estimating Student's Knowledge State via Nonlinear and Neural Attentive Models

Table 4 shows the prediction accuracy of the MAK and NAK models compared to that of the CKRM model, in terms of the RMSE and PTA metrics (**RQ2**). Both the MAK and NAK models outperform the KRM variants, with some statistically significant improvements, showing the importance of using more powerful, nonlinear models that can model the different contributions of prior courses when estimating a student' knowledge state and towards each target course. Using a maximum-based pooling layer (MAK) outperforms using an attention-based pooling layer (NAK) in the overall RMSE and PTA2, implying that the former makes less severe errors in predicting the grades.

Comparing the NAK models with the softmax and sparsemax activation functions, we can see that learning sparse attention weights outperforms learning soft attention weights. This is

Table 4: Effect of estimating students' knowledge states via nonlinear and neural attentive models.

Model	RMSE (↓)	PTA0 (↑)	PTA1 (↑)	PTA2 (↑)
KRM(sum)	0.584	32.6	70.1	87.7
KRM(avg)	0.584	34.9	70.6	87.7
MAK	0.571† (2.2)	34.7 (-0.6)	72.1 (2.1)	88.8† (1.3)
NAK(soft)	0.589 (-0.9)	35.3 (1.1)	71.8 (1.7)	88.0 (0.3)
NAK(sparse)	0.574† (1.7)	35.3† (1.1)	72.1 (2.1)	88.7† (1.1)

Note. Underlined entries represent the best performance in each metric. † denotes statistical significance over the best baseline model, using the Student's t-test with a p-level < 0.5. Numbers in parentheses denote the percentage of improvement over the best baseline value in each metric.

Table 5: Effect of modeling concurrent courses on students' performance in target courses.

Model	RMSE (↓)	PTA0 (↑)	PTA1 (↑)	PTA2 (↑)
MAK	0.571	34.7	72.1	88.8
CMAK	0.548† (4.0)	35.1† (1.2)	73.4† (1.8)	89.8 (1.1)
NAK(sparse)	0.574	35.3	72.1	88.7
CNAK	0.569† (0.9)	35.5 (0.6)	72.0 (-0.1)	88.7 (0.0)

Note. Underlined entries represent the best performance in each metric. † denotes statistical significance over the corresponding non-context-aware model, while using the Student's t-test with a p-level < 0.5.

expected, since not all prior courses are relevant to a target course, as illustrated later in the qualitative analysis in Section 9.4..

9.3. Effect of Modeling Concurrent Courses

Table 5 shows the prediction accuracy of the proposed context-aware models vs the proposed context-unaware models (**RQ3**), in terms of the RMSE and PTA metrics. CMAK outperforms MAK significantly, achieving 4% lower RMSE, and 1.1% more accurate predictions within two tick errors. On the other hand, CNAK slightly outperforms NAK with 0.9% lower RMSE, and achieves the same percentage of accurate predictions within two tick errors. This shows that modeling the interactions between a target course and concurrent courses helps in improving the prediction accuracy for a student's grade in that target course.

9.4. Qualitative Analysis of the Prior Courses Attention Weights

In this section, we study the behavior of the attention mechanism on prior courses in the NAK model (**RQ4**). Recall the motivational example for the Computer Science student, discussed in

Section 4.2.1.. This student had a set of prior courses and three target courses that we would like to predict his/her grades in (See Table 1). Using KRM(sum) or KRM(avg), all the prior courses would contribute equally to the prediction of each target course. Using our proposed NAK(sparse) model, the attention weights for the prior courses with each target course are shown in Table 6[2].

We can see that, using the sparsegen activation function, only a few prior courses are selected with non-zero attention weights, which are the most relevant to each target course.

For the Intermediate German II course, we can see that the student's grade in it is most affected by two courses: the Intermediate German I course, and the University Writing course. The Intermediate German I course is listed as a pre-requisite course for the Intermediate German II course. Though the University Writing course is not listed as a pre-requisite course, after further analysis, we found out that the Intermediate German II course requires process-writing essays and are considered part of the grading system. Though the German courses are not part of the student's degree program, and are taken by a small percentage of Computer Science students, our NAK model was able to learn accurate attention weights for them.

The other two target courses, Probability and Statistics, and Algorithms and Data Structures, have totally different prior courses with the largest attention weights, which are more related to them.

These results illustrate that the proposed NAK model was able to uncover the listed as well as the hidden/informal pre-requisite courses without any supervision given to the model.

Table 6: The attention weights of the prior courses with each target course for the sample student from Table 1.

Prior Courses	Target Course
Intermediate German I: 0.6980, University Writing: 0.3020	Intermediate German II
Calculus I: 0.4737, Physics: 0.3794, Program Design & Development: 0.0717, Operating Systems: 0.0497, Computer Networks: 0.0255	Probability & Statistics
Operating Systems: 0.2927, Advanced Programming Principles: 0.2582, Linear Algebra: 0.2313, Physics: 0.2178	Algorithms & Data Structures

Note. Prior courses are sorted in non-increasing order w.r.t. to their attention weights with each target courses for clarity purposes.

[2] These results were obtained by learning NAK models to estimate the actual grades and not the row-centered grades. Also, we used $\mathbf{q}_i = \mathbf{p}_i$ in Eq. 6. This allowed us to get more interpretable results.

10. Conclusion and Discussion

In this work, we presented context-aware nonlinear and neural attentive models that improve upon the previously developed CKRM method, by: (i) using more powerful, nonlinear models that can model the different contributions of prior courses when estimating a student's knowledge state and towards each target course; and (ii) modeling the interactions between a target course and concurrently-taken courses. The experiments showed that the proposed models significantly outperformed all baseline methods. In addition, the proposed neural attentive models are able to capture the listed as well as the hidden pre-requisite courses for the target courses, which can be better used to design better degree plans.

While CMAK and CNAK outperform existing methods, they do not take into account other contextual information, e.g., the course's instructor and the student's academic level when taking the target course. This information can further boost the accuracy of grade prediction. In addition, the proposed models assume that the course content is static and does not change over the years. Learning dynamic latent spaces for the courses could be more efficient and accurate in estimating a student's knowledge state and predicting his/her grades. Finally, since the knowledge component space is latent, these models are hard to interpret.

Acknowledgment

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

Bahdanau, D., Cho, K., and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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, J., Zhang, H., He, X., Nie, L., Liu, W., and Chua, T.-S. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 335–344.

Duchi, J., Hazan, E., and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12,* Jul, 2121–2159.

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.

He, X. and Chua, T.-S. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 355–364.

He, X., He, Z., Song, J., Liu, Z., Jiang, Y.-G., and Chua, T.-S. 2018. NAIS: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering 30,* 12, 2354–2366.

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.

KENA, G., HUSSAR, W., MCFARLAND, J., DE BREY, C., MUSU-GILLETTE, L., WANG, X., ZHANG, J., RATHBUN, A., WILKINSON-FLICKER, S., DILIBERTI, M., ET AL. 2016. The condition of education 2016. NCES 2016-144. *National Center for Education Statistics*.

LAHA, A., CHEMMENGATH, S. A., AGRAWAL, P., KHAPRA, M., SANKARANARAYANAN, K., AND RAMASWAMY, H. G. 2018. On controllable sparse alternatives to softmax. In *Advances in Neural Information Processing Systems*. 6423–6433.

MARTINS, A. AND ASTUDILLO, R. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*. 1614–1623.

MEI, L., REN, P., CHEN, Z., NIE, L., MA, J., AND NIE, J.-Y. 2018. An attentive interaction network for context-aware recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 157–166.

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.

PARIKH, A., TÄCKSTRÖM, O., DAS, D., AND USZKOREIT, J. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2249–2255.

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.

REN, Z., NING, X., AND RANGWALA, H. 2017. Grade prediction with temporal course-wise influence. In *Proceedings of the 10th International Conference on Educational Data Mining*. 48–55.

REN, Z., NING, X., AND RANGWALA, H. 2018. ALE: Additive latent effect models for grade prediction. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 477–485.

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.

VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

XIAO, J., YE, H., HE, X., ZHANG, H., WU, F., AND CHUA, T.-S. 2017. Attentional factorization machines: learning the weight of feature interactions via attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 3119–3125.