Cumulative Knowledge-based Regression Models for Next-term Grade Prediction*

Sara Morsy† George Karypis†

Abstract

Grade prediction for future courses not yet taken by students is important as it can help them and their advisers during the process of course selection as well as for designing personalized degree plans and modifying them based on the students' performance. In this paper, we present a cumulative knowledge-based regression model with different course-knowledge spaces for the task of next-term grade prediction. This method utilizes historical student-course grades as well as the information available about the courses to capture the relationships between courses in terms of the knowledge components provided by them. Our experiments on a large dataset obtained from College of Science & Engineering at University of Minnesota show that our proposed methods achieve better performance than competing methods and that these performance gains are statistically significant.

Keywords

grade prediction, regression, knowledge acquisition modeling

1 Introduction and Background

The analysis of data related to education and learning has recently gained a lot of attention by machine learning and data mining researchers. Different data mining techniques have been proposed to solve various problems in these fields, including: next-term course grade prediction [10, 13], predicting the course's final grade based on the student's on-going class performance [8,9], predicting the grades for course activities [2], knowledge tracing and student modeling [3, 7, 11], and predicting student performance in tutoring systems [4,5,12,14,15].

Many academic programs offer flexible degree plans, that include a small number of required core courses and a large number of elective courses. These electives allow students to customize their degree plans to better match their career goals. This makes course selection a crucial step that every student goes through prior to registering for each term. Our work focuses on helping students make informed decisions about which courses to register for by developing methods that can predict the grades for future courses that they have not yet taken. By knowing how well they are expected to perform in a course, students can select the courses that they are best prepared for, which can improve student retention and lead to successful and timely graduation.

A natural way to model the problem of grade prediction is to model the way the academic degree programs are structured. Each degree program requires a set of courses that need to be taken in some suggested sequence such that the knowledge provided by the earlier courses are essential for students to be able to perform well in more advanced courses. Polyzou *et al.* [10] proposed a Course-Specific Regression Model (CSRM) which builds on this idea. However, CSRM's underlying model (described in Section 4) cannot correctly capture the students' state of knowledge when the same knowledge can be acquired by taking different subsets of courses. As a result, its prediction performance deteriorates for programs with flexible degree plans.

In this paper, we develop the Cumulative Knowledge-based Regression Model (CKRM) that also builds on the idea of accumulating knowledge but addresses the aforementioned limitation of CSRM. CKRM assumes that there is a space of knowledge components describing the overall curriculum. Within that space, each course is modeled via a *knowledge component vector* that contains the knowledge components that it provides. A knowledge component can be provided by a single or multiple courses. A student by taking a course acquires its knowledge components in a way that depends on the grade that he/she obtains in that course. CKRM models the knowledge that a student has acquired after taking a set of courses via a *knowledge state vector* that is computed as the sum of the knowledge component vectors of these courses weighted by the grades that he/she has obtained in them. In order to predict the grade that a student will obtain on a specific course, CKRM estimates a per-course linear model that captures the knowledge components that are required in order to perform well in that course. Given the student's knowledge state vector prior to taking a course and that course's estimated linear model, the predicted

*This work was supported in part by NSF (IIS-0905220, OCI-1048018, CNS-1162405, IIS-1247632, IIP-1414153, IIS-1447788) 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.

†Department of Computer Science & Engineering, University of Minnesota.

grade is obtained as the dot-product of these two vectors.

We investigated three different ways of constructing the knowledge component space. Two of them construct the knowledge space in terms of an automatically identified latent space and the third uses the free text descriptions of the courses to extract keywords that form the space's dimensions. The difference between the two latent spaces is that one imposes the constraint that courses from different departments do not share any knowledge components, whereas the other one does not.

The main contributions of this work are as follows:

1. We propose a cumulative knowledge-based method for the problem of next-term grade prediction that better models the structure of degree programs and is better suited for flexible degree programs.

2. We performed an extensive experimental evaluation on a real world dataset containing 10 years worth of student grades from 12 academic departments from the College of Science and Engineering at University of Minnesota. This evaluation showed that the proposed methods perform statistically significantly better than competing approaches.

3. We showed that the models that were estimated based on the extracted keywords can identify the knowledge that is required in order to perform well in a course, which is not captured by the course pre-requisites. This can be used to inform changes in course sequencing and degree programs.

The rest of the paper is organized as follows. Section 2 explains the definitions and notations used throughout the paper. We explain our proposed methods in Section 3. Section 4 reviews the previous research that is relevant to our proposed methods. Section 5 describes the experimental setup and evaluation methodology. The results are shown in Section 6. Finally, we conclude our work in Section 7.

2 Notations and Definitions

Boldface uppercase letters will be used to represent matrices (e.g., $\mathbf{G}, \mathbf{R}$) and boldface lowercase letters to represent row vectors, (e.g., $\mathbf{r}$). The ith row of matrix $\mathbf{A}$ is represented as $\mathbf{a}_i$. The entry in the ith row and jth column of matrix $\mathbf{A}$ is denoted as $a_{i,j}$. 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.

For a student s and a course c not yet taken by s and he/she would like to register for it in the following term, we define the problem of predicting the grade that s will obtain on c as the *next-term grade prediction* problem, or the *grade prediction* problem for short.

3 CKRM: Cumulative Knowledge-based Regression Models

Consider a student s that has taken j courses $\langle c_1, \ldots, c_j \rangle$ in that sequence, and a course c that s has not yet taken for which we will like to predict his/her grade. A course c is assumed to provide a set of knowledge components that the student acquires after taking c. These knowledge components can be the set of topics or concepts taught by the course. We assume that all courses can be represented in a knowledge space of these different components. We will refer to the knowledge component vector of a course c as its *provided knowledge component vector* and we will denote it as $\mathbf{p}_c$. We define the *knowledge state* for student s after taking j courses as the knowledge he/she has acquired so far in the different knowledge components provided by the j courses. A student's s knowledge state after taking j courses will be denoted by the *knowledge state vector* $\mathbf{k}_{s,j}$ and will be computed as

$$\mathbf{k}_{s,j} = \sum_{i=1}^{j} \Big(\xi(s, c_j, c_i)\ g_{s,c_i}\ \mathbf{p}_{c_i} \Big), \tag{3.1}$$

where g_{s,c_i} is the grade that student s obtained on course c_i, and $\xi(s, c_j, c_i)$ is a time-based exponential decaying function designed to de-emphasize courses that were taken a long time ago. This equation models a student's knowledge state as the sum of the provided knowledge component vectors of the courses he/she has taken so far, weighted by his/her grades in them. The grade-based weighting is designed to capture the fact that a student better acquires the knowledge components of a course on which he/she obtained a good grade than a course on which he/she did not.

The decaying function that we used is:

$$\xi(s, c_j, c_i) = e^{-\lambda(t_{s,c_j} - t_{s,c_i})}, \tag{3.2}$$

where λ is a user-specified non-negative parameter that controls the shape of the exponential decaying function, and t_{s,c_i} is the term number when student s took course c_i. This term number is encoded as follows. For each student, we encode his/her first term as the term numbered as 1, and each following term number is incremented by 1. This numbering technique applies a time-based decaying importance on the prior courses, such that as the time difference between taking course c_i and the most recent taken course c_j increases, the effect of the corresponding components of c_i (namely, $g_{s,c_i}\mathbf{p}_{c_i}$) on the student's current knowledge state (after taking j courses) will get smaller.

CKRM computes the grade that student s will obtain on a course c by applying a course-specific linear model $\mathbf{r}_c$ on the student's knowledge state vector prior to taking c. That is, the predicted grade is given by

$$\hat{g}_{s,c} = \mathbf{r}_c \, \mathbf{k}_{s,j}^T, \tag{3.3}$$

where $\mathbf{k}_{s,j}$ is the corresponding knowledge state vector. These course-specific linear models are estimated from the historical grade data and can be considered as capturing and weighting the knowledge components that a student needs to have accumulated in order to perform well in a course. For this reason, we will refer to these linear models as the *required knowledge component vectors*.

3.1 The course knowledge component space. In order to capture the knowledge components provided by courses, we experiment with three different ways of defining the structure of the knowledge component space. Two of them are based on a latent space, and the third one is based on the textual descriptions of these courses.

3.1.1 Latent knowledge component space. The most straightforward way to define the latent knowledge component space is to use the standard latent structure in which all dimensions, i.e., knowledge components, are shared across all courses. We will refer to the CKRM-based method that uses the standard latent structure as **CKRMall**. For academic courses that belong to different departments, however, they should not share their provided knowledge components among each other. For instance, a course that belongs to Mechanical Engineering in general should not share any of its provided knowledge components with a course from Computer Science & Engineering.

In order to model this, we experiment with a "prescribed" latent structure, which is based on the assumption that courses belonging to the same department provide the same set of knowledge components and that courses belonging to different departments do not share any of their provided knowledge components with each other. In this case, we allocate a distinct set of l latent dimensions for each department. For example, if $l = 5$, and we are working with 10 departments, then the number of dimensions for that approach will be $5 \times 10 = 50$ dimensions. We will refer to the CKRM-based method that uses this prescribed latent structure as **CKRMdep**.

Within that prescribed structure, for each provided knowledge component vector ($\mathbf{p}_c$) we will need to estimate only l values, whereas for each required knowledge component vector ($\mathbf{r}_c$), we can potentially be estimating all dimensions.

3.1.2 Textual-based knowledge component space. A source that offers information about the knowledge components provided by courses is their textual descriptions in the University course catalog. These are usually short descriptions of what different knowledge components are provided by the courses in a form of free-text sentences and/or keywords. We hypothesize that it may be possible to derive a knowledge component space using these descriptions.

In order to test this hypothesis, we will use the set of 2-ngrams that co-exist within a specific window in the textual descriptions of the courses as the knowledge component space and represent each course as a bag-of-ngrams vector. With this representation, we can use the vectors in the knowledge component space as indicator vectors and just estimate the required knowledge component space, or we can estimate the non-zero entries of the provided knowledge component space along with estimating the required knowledge component space. In the latter case, the weights on the provided knowledge component vectors can be viewed as indicating some type of relative importance of the different dimensions (i.e., ngrams) in that course. We will refer to the CKRM-based method that uses the textual descriptions of courses as **CKRMtext**.

3.2 Parameter estimation. The parameters of the CKRM-based methods are the required knowledge component vectors associated with each course, i.e., the various $\mathbf{r}_c$ vectors, and the provided knowledge component vectors of each course, i.e., the $\mathbf{p}_c$ vectors (the latter vectors are estimated for all the approaches except when using them as indicator vectors in CKRMtext).

We use the squared error loss function to estimate these parameters. For the approaches that estimate the provided knowledge component vectors, the optimization problem is

$$\begin{aligned} \underset{\mathbf{R},\mathbf{P}}{\text{minimize}} \; & \frac{1}{2} \sum_{s,c \in \mathbf{G}} (g_{s,c} - \hat{g}_{s,c})^2 + \frac{\alpha}{2} \Big(\|\mathbf{R}\|_F^2 + \|\mathbf{P}\|_F^2 \Big) \\ \text{subject to } & \mathbf{R} \geq 0, \; \mathbf{P} \geq 0, \end{aligned} \tag{3.4}$$

where $g_{s,c}$ is the actual grade, $\hat{g}_{s,c}$ is the predicted grade (computed as in Eq. 3.3), $\mathbf{R} \in \mathbb{R}^{n \times d}$ is the matrix whose rows are the required knowledge component vectors, $\mathbf{P} \in \mathbb{R}^{n \times d}$ is the matrix whose rows are the provided knowledge component vectors, and α is a regularization parameter to avoid overfitting. The non-negativity constraints on $\mathbf{R}$ and $\mathbf{P}$ are enforced since they represent knowledge acquisition, which should be non-negative. Note that for CKRMdep and CKRMtext, $\mathbf{P}$ has a

predefined sparse structure, so only the weights of its encoded non-zero entries are estimated. For CKRMtext that uses the provided knowledge component vectors as indicator vectors, the optimization problem is

$$\underset{\mathbf{R}}{\text{minimize}} \ \frac{1}{2} \sum_{s,c \in \mathbf{G}} (g_{s,c} - \hat{g}_{s,c})^2 + \frac{\alpha}{2} \|\mathbf{R}\|_F^2$$

$$\text{subject to } \mathbf{R} \geq 0. \tag{3.5}$$

The optimization problems of Eqs. 3.4 and 3.5 are solved using a Stochastic Gradient Descent (SGD) algorithm, which is an iterative algorithm. Algorithm 1 provides the detailed procedure and gradient update rules. Matrices $\mathbf{R}$ and $\mathbf{P}$ are initialized with small random values as the initial estimate (line 6). In each iteration of SGD (lines 7–25), if the course has at least l courses taken prior to it, then its required knowledge component vector $\mathbf{r}_c$ is updated as well as the preceding j courses' provided knowledge component vectors $\mathbf{p}_{c_i}$. This process is repeated until the RMSE on the validation set does not decrease further or the number of iterations has reached a predefined threshold. Note that, for solving Eq. 3.5, lines 18–21 are ignored and the non-zero entries of $\mathbf{P}$ are just used as indicator vectors.

Algorithm 1 CKRM:Learn

1: **procedure** CKRM_Learn
2: $l \leftarrow$ minimum # prior courses
3: $\eta \leftarrow$ learning rate
4: $\alpha \leftarrow$ regularization weight
5: $iter \leftarrow 0$
6: Init the non-zero entries of $\mathbf{R}$ and $\mathbf{P}$ with random values in [-0.001, 0.001]
7: **while** $iter < maxIter$ or RMSE on validation set decreases **do**
8: **for all** $g_{s,c} \in \mathbf{G}$ **do**
9: $j \leftarrow$ # courses taken by s prior to c
10: **if** $j \geq l$ **then**
11: $c_j \leftarrow$ last course taken by s prior to c
12: $\mathbf{k}_{s,j} \leftarrow 0$
13: **for all** $c_i \in \mathbf{g}_s$ s.t. c_i was taken by s prior to c **do**
14: $\mathbf{k}_{s,j} \leftarrow \mathbf{k}_{s,j} + \xi(s, c_j, c_i)\ g_{s,c_i}\ \mathbf{p}_{c_i}$
15: **end for**
16: $\hat{g}_{s,c} \leftarrow \mathbf{r}_c\ \mathbf{k}_{s,j}^T$
17: $e_{sc} \leftarrow g_{s,c} - \hat{g}_{s,c}$
18: $\mathbf{r}_c \leftarrow \mathbf{r}_c + \eta \cdot (e_{s,c} \cdot \mathbf{k}_{s,j} - \alpha \cdot \mathbf{r}_c)$
19: **for all** $c_i \in \mathbf{g}_s$ s.t. c_i was taken by s prior to c **do**
20: $\mathbf{p}_{c_i} \leftarrow \mathbf{p}_{c_i} + \eta \cdot (e_{s,c} \cdot g_{s,c_i} \cdot \mathbf{r}_c - \alpha \cdot \mathbf{p}_{c_i})$
21: **end for**
22: **end if**
23: **end for**
24: $iter \leftarrow iter + 1$
25: **end while**
26: **return** $\mathbf{R}$ and $\mathbf{P}$
27: **end procedure**

4 Review of Relevant Research

In recent years different techniques have been proposed for solving the next term grade prediction problem. The majority of these methods leverage ideas that were developed in the context of recommender systems [1, 10, 13] as well as approaches that are based on standard classification and regression [10, 13].

One of the approaches proposed by Polyzou et. al. [10] is a cumulative knowledge-based model, called Course-specific Regression Model (CSRM), which is based on the fact that the student's performance in a future course is based on his performance in the past courses. Consider a student s that has taken j courses $\langle c_1, \ldots, c_j \rangle$ in that sequence, and a course c that s has not yet taken for which we will like to predict his/her grade. In CSRM, the grade for student s in course c is predicted as a sparse linear combination of his previous grades, which is computed as

$$\hat{g}_{s,c} = \mathbf{r}_c^T \Big(\sum_{i=1}^{j} g_{s,c_i} \mathbf{z}_{c_i} \Big), \tag{4.6}$$

where $\mathbf{r}$ and $\mathbf{z}$ are vectors of dimension equal to the total number of courses n, $\mathbf{r}_c$ is a linear model associated with course c, g_{s,c_i} is the grade that student s obtained on course c_i, and $\mathbf{z}_{c_i}$ is an indicator vector with one in the dimension corresponding to course c_i.

Since CSRM treats each course as having a unique dimension that does not share anything with any other course, it assumes that each course provides a set of knowledge components that are totally different from any other course, which does not hold for many courses. As we will see in the experimental results (Section 6), the capability of CSRM to accurately model the accumulation of knowledge decreases as the flexibility of the degree program increases, i.e., as students can take more diverse courses that provide the same or similar knowledge components prior to taking the target course.

Similar models to CKRM have also been explored in the context of recommender systems, in which models are developed for item rating prediction, such as the factored item-similarity model (FISM) presented in [6]. The difference between the two methods is that, in CKRM, there are temporal dependencies, in which the prediction for the grade that student s will obtain on

course c depends on the courses that s has taken prior to c, unlike FISM, which aggregates over all items without taking the rating time into account.

5 Experimental Evaluation

5.1 Dataset description and preprocessing. The data used in our experiments was obtained from the College of Science and Engineering at University of Minnesota and includes 12 degree programs. The data that we used span a period of about 10 years (Fall 2006 to Spring 2015). From that dataset, we extracted the students who were registered at the University for at least three terms[1]. For each of these students, we extracted the set of courses that belong to these 12 majors. We removed any courses that were taken as pass/fail. The initial grades were in the A–F scale, which were converted to the 4–0 scale using the standard letter grade to GPA conversion. If a course was taken more than once by a student, then only its most recent grade is retained and the older ones are eliminated. The statistics of the extracted majors are shown in Table 1.

Table 1: Information about the different majors.

Major	Abbrev.	#Students	Flexibility
Mathematics	MATH	1,032	0.704
Statistics	STAT	289	0.698
Physics	PHYS	241	0.664
Chemistry	CHEM	665	0.653
Computer Science	CSE	1,293	0.609
Electrical Eng.	ECE	737	0.589
Materials Science	MATS	272	0.520
Chemical Eng.	CHEN	785	0.512
Mechanical Eng.	ME	1,302	0.490
Biomedical Eng.	BMEN	524	0.485
Aerospace Eng.	AEM	450	0.439
Civil Eng.	CE	560	0.439

The majors are sorted with respect to their flexibility in a decreasing order (see Section 5.1 for the definition of the major's flexibility).

Table 2: Datasets statistics.

	Train	Validation	Test
#Students	59,054	27,101	21,797
#Courses	8,708	3,941	1,318
#Grades	856,025	83,518	56,915

These statistics are accumulated over the eight datasets created for the eight test terms (see Section 5.2).

Table 1 also shows each major's *flexibility*, which is a measure that we computed in order to differentiate between degree programs that that have a large number of electives and the students' degree programs tend to include different sets of courses (flexible) over those that offer a few electives and the degree programs of all students are quite similar (restricted). As our results will show, the major's flexibility impacts the performance of certain models. We computed the major's flexibility as the average course offering flexibility over all course offerings that belong to that major, weighted by the number of pairs of students in that offering. We computed the flexibility of a course offering c as one minus the average Jaccard coefficient of the courses that were taken by the students that took c prior to taking this class. The flexibility will be low if the students that took c have taken very similar courses before c and high otherwise. Note that only the students belonging to each major were used while computing its flexibility.

Since the CKRM-based methods rely on extracting the different knowledge areas/components provided by the courses, we manually removed courses that do not provide consistent knowledge, such as independent study, directed research, and other non-topic specific courses. For CKRMtext, we extracted the 2-ngrams from each course description that exist within a window of size three after removing the stopwords and created a course-by-ngram matrix that was used as the provided knowledge component matrix $\mathbf{P}$.

5.2 Generating train, validation, and test sets. The entire dataset was used to extract eight different subsets in order to assess the performance of the different methods. Specifically, we selected the eight most recent Fall and Spring terms in our dataset to predict their grades (which we will refer to as the set of test terms $\mathcal{T}$), where for each of these test terms $t \in \mathcal{T}$, only the terms prior to t are used for training and validation. The training, validation and test sets were extracted as follows. For each test term t, the term prior to it that is either a Fall or a Spring term (not a Summer term) is used for validation and model selection, and all the terms prior to the validation term are used for learning the model. For a student to be considered in the training set, he/she must have taken at least three courses in the training set. This is to ensure that the students have taken a sufficient number of courses so that CKRM can capture knowledge accumulation. Also, we did not consider a course for predicting its grades in the validation or test set if its required knowledge component vector ($\mathbf{r}_c$) was estimated, during learning the model, less than 50 times, as we considered such courses not to have reliable estimated required knowledge component vectors. Therefore, for a course to be considered for prediction during validation or testing, it must have been taken by at least 50 students after at least 3 courses. The statistics about the accumulated training, validation and test sets over the eight subsets of data are shown in Table 2.

[1] There are three terms at this University: Fall, Spring and Summer.

Table 3: Prediction performance of the different methods on major students.

#Ticks	Method	MATH	STAT	PHYS	CHEM	CSE	ECE	MATS	CHEN	ME	BMEN	AEM	CE
Percentage of grades predicted with no error	BiasOnly	18.41	18.99	23.20	20.01	22.13	23.90	20.14	17.51	24.55	26.73	28.12	23.12
	MF	18.65	18.85	24.18	21.26	22.88	24.63	21.94	18.85	24.40	26.52	26.56	22.88
	CSRM	18.52	16.83	22.83	19.94	23.54	26.20	19.43	21.51	25.65	29.28	30.34	23.33
	CKRMdep	19.28	18.70	23.20	20.94	24.41	26.14	22.01‡	20.98†	26.06	29.24	30.41	23.57
	CKRMall	19.21	17.84	24.30	21.26	24.63	26.33	20.98‡	21.15†	26.36‡	29.58	31.01†	23.93
	CKRMtext	19.22	18.42	25.28	21.45	25.03†	26.09	22.84‡	21.72†	26.20	30.13	30.76†	23.62
Percentage of grades predicted with an error of at most one tick	BiasOnly	52.65	55.11	60.93	52.88	59.31	63.51	53.60	54.42	63.17	69.15	68.36	60.80
	MF	52.85	54.39	61.30	53.33	60.50	63.94	55.92	55.79	63.49	71.01	69.73	61.23
	CSRM	51.62	54.10	58.61	54.20	61.98	64.15	57.27	58.94	65.78	73.69	71.65	63.87
	CKRMdep	54.06‡	56.55	60.32	54.78	62.84†	64.58‡	57.47	59.10†	65.90†	74.12†	71.61†	64.20
	CKRMall	54.10‡	56.54	60.81	54.90	63.05†‡	64.41‡	57.15	58.74†	66.00†	74.84	71.44†	64.61
	CKRMtext	53.99‡	56.40	59.95	54.71	63.24†	64.24‡	59.27†‡	59.19†	65.97†	74.67†	71.44†	64.56
Percentage of grades predicted with an error of at most two ticks	BiasOnly	76.96	76.84	83.64	75.85	82.49	84.82	78.05	80.41	85.63	90.36	89.45	83.61
	MF	76.99	76.98	84.25	76.17	82.67	84.67	79.54	80.58	85.70	90.44	90.04	83.68
	CSRM	76.68	80.86	81.93	77.00	83.89	84.43	81.72	82.65	86.12	91.50	87.67	84.71
	CKRMdep	77.22	81.58†	83.39	76.75	84.61†‡	84.70	79.86	83.26†‡	85.81	91.80	87.85	84.87†
	CKRMall	77.55	82.01†	83.76	76.65	84.72†‡	84.55	80.76	82.66†	85.98	91.50	88.06	85.09†
	CKRMtext	77.59	81.30†	82.29	76.55	84.40†‡	84.32	80.63	83.09†	85.98	91.54	87.99	85.21†
# Predicted Grades		4,632	695	819	3,109	7,180	5,344	1,554	5,305	7,118	2,353	2,864	4,179

The majors are sorted in descending order with respect to their flexibility (see Table 1). See Section 5.5 for the definition of a tick. Underlined entries denote the best value obtained for each major for each #ticks. † denotes statistical significance over the best of MF and BiasOnly, whereas ‡ denotes statistical significance over CSRM, both at the 5% significance level. The parameters for the selected models are shown in the Appendix.

5.3 Grade standardization. A characteristic of the course grade data is that the mean and standard deviation of the same course vary across its different offerings, for many reasons, including the instructor who teaches the course, the knowledge states and background skills of the students taking the class, and the structure of the course evaluation method. In order to deal with such course offering variations, we used the standardized z-scores that are computed on a per-course-offering basis, as follows. We first computed the global standard deviation using all the grades in the training set. Then, for each course offering, we computed the mean and standard deviation of its grades. We then computed the z-score for a grade g in a course offering as

$$z = \frac{g - \mu}{\sigma_{local} + \sigma_{global}}, \tag{5.7}$$

where μ and σ_{local} are the mean and standard deviation of the grades for that course offering, respectively, and σ_{global} is the global standard deviation. We added σ_{global} as a damping factor in order to widen the range of the set of the computed z-scores. Since these z-scores are not restrictively non-negative, we removed the constraint of non-negativity on $\mathbf{R}$ while estimating the CKRM-based methods using the z-scores.

After estimating the parameters of the model, we converted the predicted (standardized) grades into actual grades using the information on the past course offerings. Specifically, for a course c for which we predicted its grades, we approximated its mean as the mean of the means of the grades of its past offerings, and its standard deviation as the sum of the global standard deviation and the mean of the standard deviations of the grades of its past offerings. Note that this approach does not use any information about the distribution of the grades of the test course and as such, the predicted grades represent predictions for a course that has not been yet completed.

5.4 Baseline/Competing methods. In our experiments, we compared the performance of the CKRM-based methods against the following competing methods:

1. **CSRM:** This is the course specific regression model that was described in Section 4.
2. **Matrix Factorization (MF):** This approach predicts the grade for student s in a course $c$as

$$\hat{g}_{s,c} = \mu + sb_s + cb_c + \mathbf{u}^{\mathbf{T}}\mathbf{v}, \tag{5.8}$$

where μ, sb_s and cb_c are the global, student and course bias terms, respectively, and $\mathbf{u}$ and $\mathbf{v}$ are

Table 4: Prediction performance of the different methods on non-major students.

#Ticks	Method	MATH	STAT	PHYS	CHEM	CSE	ECE	MATS	ME	AEM	CE
Percentage of grades predicted with no error	BiasOnly	18.19	24.51	18.82	19.20	20.59	19.16	19.77	16.09	21.45	13.90
	MF	19.55	27.54	18.99	19.73	21.93	21.41	19.07	17.69	21.25	17.76
	CSRM	20.99	17.29	18.98	21.14	18.69	22.23	24.29	18.23	21.25	14.67
	CKRMdep	20.69	20.95†	17.68‡	22.85	18.61†	23.56	23.45†	16.35	20.51	18.53
	CKRMall	20.27	21.92†	18.82‡	23.16	20.90	24.69	23.45†	18.50	22.12	15.06
	CKRMtext	21.48	19.52†	19.15	23.51	19.87†	25.00	22.32†	16.62	21.92	18.92
Percentage of grades predicted with an error of at most one tick	BiasOnly	53.83	62.84	56.63	55.66	58.04	56.76	55.22	47.72	56.35	49.42
	MF	53.79	63.01	56.30	56.27	58.19	56.15	57.91	50.40	56.55	49.81
	CSRM	53.61	56.59	54.99	57.55	55.58	57.99	57.48	57.64	56.09	42.08
	CKRMdep	53.72	59.18‡	54.50	60.85	56.29	59.32	58.19	56.57	55.82	47.50‡
	CKRMall	54.10	59.27‡	55.48	60.77†	56.93	60.04	59.46	56.30†	57.23	47.11
	CKRMtext	53.91	59.80	54.66	61.47†‡	55.74†	60.55	60.31	54.16†‡	56.42	43.25
Percentage of grades predicted with an error of at most two ticks	BiasOnly	76.21	83.96	81.51	79.79	79.10	81.24	82.20	78.55	78.48	78.77
	MF	76.03	83.87	81.01	80.14	77.91	81.14	82.77	79.09	78.68	76.84
	CSRM	75.39	82.44	79.05	80.58	80.05	82.37	83.61	79.62	78.89	72.97
	CKRMdep	75.96	82.62	79.05	82.78	77.67‡	82.07	84.89	81.50	79.02	73.36
	CKRMall	76.22	82.09†	80.19	82.48	77.36‡	81.96‡	84.89	81.77	79.35	75.68
	CKRMtext	76.26†‡	81.82	79.54	82.74‡	77.83‡	81.35	85.03	82.30	79.55	74.52
# Predicted Grades		2,649	1,122	611	2,271	1,263	976	708	373	1,487	259

Underlined entries denote the best value obtained for each major for each #ticks. † denotes statistical significance over the best of MF and BiasOnly, whereas ‡ denotes statistical significance over CSRM, both at the 5% significance level.

the student and course latent vectors, respectively.

We used the squared loss function with L2 regularization to estimate this model, by solving the following optimization problem:

$$\underset{\mu,\mathbf{sb},\mathbf{cb},\mathbf{U},\mathbf{V}}{\text{minimize}} \ \frac{1}{2}\sum_{s,c\in\mathbf{G}}(g_{s,c}-\hat{g}_{s,c})^2+\frac{\alpha}{2}\Big(\|\mathbf{sb}\|_2^2 + \|\mathbf{cb}\|_2^2+\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2\Big), \tag{5.9}$$

where: **sb** and **cb** are the student and course bias vectors, respectively, and **U** and **V** are the student and course latent factor matrices, respectively.

3. **BiasOnly:** This method is a special case of MF, in which the number of latent dimensions is 0. That is, it predicts the grade for student s in a specific course c using only the bias terms in Eq. 5.8.

5.5 Evaluation methodology and performance metrics. We evaluated the performance of the different approaches by using them to predict the grades for each of the eight test terms in our dataset using the data from the terms prior to each test term for training and validation (see Table 2).

The grading system used by the University 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 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. We converted the predicted grades into their closest letter grades and then computed the percentages of each of the x ticks, where the number of x-ticks denotes the number of predicted grades that were x ticks away from their actual grades.

We believe that the grades that are predicted with at most one or two ticks error are sufficiently accurate for the task of course selection and that the grades that can be predicted with an error of three or more ticks can incorrectly influence course selection. For these reasons, we will tend to refer to the grades that are predicted with at most one or two ticks error to be *sufficiently accurate*.

We also performed statistical significance tests of the CKRM-based methods against the competing methods as well as against each other. We ran a paired-sample one-tailed t-test using the ticks percentages of

Table 5: Effect of major's flexibility on the performance of the CKRM-based methods against the competing methods on major and non-major students.

	Average Percentage Improvement					
#Ticks	**Most Flexible**	**Flexible**	**Least Flexible**	**Most Flexible**	**Flexible**	**Least Flexible**
	Major Students			Non-major Students		
Baseline	CSRM					
No error	8.37	6.34	2.60	10.31	6.95	11.52
Within one tick	3.60	1.65	0.75	3.57	3.92	4.35
Within two ticks	1.13	0.21	0.30	1.39	-0.48	2.64
Baseline	Best of MF & BiasOnly					
No error	1.83	8.90	8.47	2.38	10.21	4.74
Within one tick	1.78	4.40	4.39	0.65	2.89	2.93
Within two ticks	1.88	1.80	0.37	0.04	0.71	0.42

The 12 majors are divided into three groups of four majors each, according to their flexibility (see Table 1). Each of these percentages is averaged over the included majors' percentage improvements.

the courses belonging to each major in each of the eight datasets as the data points for each method.

5.6 Model selection. We did an extensive search in the parameter space for model selection. We experimented with the regularization parameter α in the range [1e-5, 0.1] and with the learning rate η in the range [5e-5,1]. For CKRMall and CKRMdep, we used the number of dimensions: {10, 20, 30}, whereas for MF we used it in the range [10,60] with a step of 5. For the CKRM-based methods, we experimented with the parameter λ in the range [0, 1] with a step of 0.1.

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 RMSE of the validation set.

6 Experimental Results

For each of the 12 departments, we divided the results into the set of students that belong to the same department (*major* students) and the set of students that belong to one of the remaining 11 departments (*non-major* students), since these two groups of students represent different populations for each department.

We organized the experimental results into three parts. The first and second show a quantitative comparison of the CKRM-based methods against each other as well as against the competing methods on major and non-major students, respectively. Finally, the third shows a qualitative analysis on CKRMtext.

6.1 Quantitative performance on major students. Table 3 shows the performance achieved by the CKRM-based and competing methods on major students in terms of the percentage of grades predicted with no error, with an error of at most one tick, and with an error of at most two ticks.

Comparing the performance achieved by the three CKRM-based methods, we can see that their performance is quite similar. If we consider the best performing entries across the different departments and error levels we see that one of them outperforms the other two. However, even when a method does better than another one, the differences are fairly small. The close performance of the three methods was also confirmed by the statistical significance tests that we ran, which showed that the performance difference of the three schemes were not statistically significant for most departments.

Comparing the performance achieved by the CKRM-based methods against that achieved by CSRM, we see that the former leads to more accurate predictions and its performance advantage is greater for the flexible majors than the restricted ones. This is further illustrated in Table 5, which shows the average percentage improvements of the CKRM-based methods based on the majors' flexibility. The CKRM-based methods achieve an average improvement of 8.37, 3.6, and 1.13% over CSRM in the most flexible majors, as opposed to 2.6, 0.75, and 0.3% in the least flexible ones for the no error, within one tick, and within two ticks errors, respectively. These percentage improvements also indicate that the CKRM-based methods do considerably better than CSRM in terms of the no error predictions. These results confirm our hypothesis that CSRM's performance degrades as the major's flexibility increases, since this method depends on the prior set of courses to predict the grades, which can fail in such flexible majors as each student can take a different combination of courses that offer the same knowledge components required for performing well in that course.

Comparing the performance achieved by the CKRM-based methods against that achieved by MF and BiasOnly, we see that they also outperform both MF and BiasOnly in most cases and that their performance

is statistically significant over both baselines in some cases. As shown in Table 5, the CKRM-based methods tend to have greater improvement over MF and BiasOnly in the flexible and least flexible major groups than in the most flexible ones.

Comparing the performance of CSRM against that of both MF and BiasOnly, we can see that CSRM does generally better in the less flexible majors and worse in the more flexible ones, as it is more suited to the less flexible majors, as we explained above.

6.2 Quantitative performance on non-major students. Table 4 shows the prediction performance achieved by the CKRM-based and competing methods on the set of non-major students in terms of the percentage of grades predicted with no error, with an error of at most one tick, and with an error of at most two ticks.

Comparing the performance achieved by the three CKRM-based methods, we can see that their performance is quite similar, and there was no statistically significant difference in their performance. Comparing the performance of the CKRM-based methods against that of the competing approaches, we can see that the former lead to more accurate predictions that are statistically significant in most departments. Both MF and BiasOnly tend to outperform the other methods for the prediction of the STAT grades.

The last three columns of table 5 show the average percentage improvements of the CKRM-based methods based on the majors' flexibility[2]. As shown in the table, the performance of the CKRM-based methods leads to more accurate predictions than the competing approaches, and they do considerably better in terms of the no error predictions.

6.3 Qualitative analysis of CKRMtext's models. The fact that the performance of CKRMtext's models are comparable to that of the other two latent space based variants of CKRM (as discussed in Section 6.1) is important, because the models estimated by CKRMtext are easier to interpret (since their dimensions correspond to keywords extracted from the course descriptions). As a result, they can be analyzed in order to learn, from students' historical data, the importance of each of the knowledge components for each course.

For this reason, we analyzed the results of CKRMtext's models, as follows. For each course, we extracted, from the students who took that course, the top 2-ngrams that have the highest weights in their knowledge states prior to taking that course (see Eq. 3.1) and computed the percentage of its extracted top ngrams matching the descriptions of the course's pre-requisites[3].We found that most courses have their top ngrams matching only 0–39% of their pre-requisite descriptions. This suggests that there are other knowledge components not listed in the course's pre-requisite descriptions that also affect the student's performance in that course.

In order to better understand the type of information that these "other" knowledge components capture, we manually analyzed the top-20 ngrams for the CSE courses. Table 6 shows a sample of four of these courses along with their top ngrams. We can see that the ngrams (shown in black) that are not included in the text description of the pre-requisites are also relevant for the requirements of these courses. For instance, for the network course (CSCI 5221), there are three ngrams that contain the word "java" ("java object", "java programming" and "java oriented"), along with other ngrams about programming languages in general. This suggests that the students' performance in the programming courses, especially those that taught the Java language had significant impact on their performance in that course. Another example is the Artificial Intelligence course (CSCI 5512), which has eight of its top 20 ngrams, namely "control programming", "applications sensing", "dynamics kinematics", "applications programming", "based programming", "applications based", "inverse kinematics", and "applications robotics", not appearing in the pre-requisites. However, after some further analysis, we determined that these ngrams appear in the description of the CSE course entitled "CSCI 5551, Introduction to Intelligent Robotic Systems", which is not listed as a pre-requisite for that course. This also suggests that students' performance in CSCI 5551 along with the other introductory CSE courses that contain the remaining top ngrams highly affect their performance in CSCI 5512. Similar insights can be gained from the other courses.

This analysis can provide information about the "hidden" or "informal" knowledge components whose acquisition by previous students have greatly affected their performance in the target courses. Moreover, these knowledge components can be mapped back to their corresponding courses, which would tell us about the specific courses that have more impact on the performance of students in these courses. This can help in improving the pre-requisite structure and/or the suggested degree plans of the various degree programs

[2]Note that the major's flexibility are based on major students only

[3]These results were obtained by learning models to estimate the actual grades and not the z-scores. This allowed us to have both $\mathbf{R}$ and $\mathbf{P}$ to be non-negative and as such made the results more interpretable.

Table 6: Top-20 textual features for a sample of four CSE courses.

CSCI 2011 – Discrete Structures of Computer Science
Top Features: calculus space:15.97, functions polynomials:12.72, quantitative systems:9.76, integration involving:9.48, principles systems:9.21, introduction programming:8.63, language languages:8.26, curves space:8.23, language structures:8.15, data languages:7.8, functions taylor:7.79, calculus integration:7.62, language programming:7.5, data programming:6.44, involving taylor:6.25, forces mechanical:6.24, modularity programming:6.2, languages programming:6.03, development program:5.58, motion systems:5.53
CSCI 4203 – Computer Architecture
Top Features: logical models:6.38, analysis models:4.91, computer machine:4.35, languages models:4.24, mathematical models:2.48, data languages:2.25, computer mathematical:2.17, computer programming:1.98, introduction programming:1.76, probability sampling:1.69, analysis data:1.67, formal models:1.63, computer models:1.61, distributions sampling:1.38, functions methods:1.27, networks programming:1.16, programming projects:1.13, algebra boolean:1.11, communication projects:1.1, development program:1.06.
CSCI 5221 – Foundations of Advanced Networking
Top Features: data programming:3.12, data network:2.94, computer programming:2.21, language structures:1.69, networks programming:1.61, language programming:1.21, architectures routing:1.1, architectures examples:1.06, development program:1.05, computer science:0.95, java object:0.94, network programming:0.92, architectures protocols:0.81, java programming:0.72, communication programming:0.68, architectures network:0.68, computer data:0.64, data networks:0.62, concepts programming:0.6, java oriented:0.54.
CSCI 5512 – Artificial Intelligence II
Top Features: language structures:1.44, computer programming:1.37, data programming:1.24, introduction programming:1.16, control programming:1.16, computer machine:1.13, language programming:1.06, applications sensing:1.04, analysis data:1.01, dynamics kinematics:0.98, java object:0.95, introduction theorem:0.92, applications programming:0.89, based programming:0.87, differential equations:0.85, applications based:0.82, analysis design:0.79, inverse kinematics:0.73, development program:0.72, applications robotics:0.67.

The ngrams colored in red denote those that exist in the course's pre-requisite descriptions. The weight of each ngram is shown next to it, which is computed as explained in Section 6.3.

in order to take the actual "learned" structure into account. It can also help in providing future students with the knowledge components (or courses) that have had more impact on the previous students' performance in the different courses, other than the ones listed in the course's pre-requisites.

7 Conclusion

In this paper, we modeled the next-term grade prediction problem in a traditional University setting as a Cumulative Knowledge-based Regression Model (CKRM) that accumulates the performance of a student in all the courses that he/she has previously taken in order to predict his/her future grades. We conducted an extensive experimental evaluation on a large dataset that includes 12 degree programs of the College of Science & Engineering at University of Minnesota. The results showed that the CKRM-based methods are able to estimate more accurate predictions than the competing methods. Moreover, the analysis of the CKRM-based methods that use the textual course descriptions showed that they can be used to identify the knowledge required for students to perform well in courses.

References

[1] Bydžovská *et al.* Are collaborative filtering methods suitable for student performance prediction? In *Progress in Artificial Intelligence*. 2015.

[2] Elbadrawy *et al.* Collaborative multi-regression models for predicting students' performance in course activities. In *LAK*, 2015.

[3] González-Brenes *et al.* Dynamic cognitive tracing: Towards unified discovery of student and cognitive models. *EDM*, 2012.

[4] Hershkovitz *et al.* Predicting future learning better using quantitative analysis of moment-by-moment learning. In *EDM*, 2013.

[5] Hwang *et al.* Unified clustering locality preserving matrix factorization for student performance prediction. *IAENG Int. J. Comput. Sci*, 2015.

[6] Kabbur *et al.* Fism: Factored item similarity models for top-n recommender systems. In *KDD*, 2013.

[7] Lan *et al.* Sparse factor analysis for learning and content analytics. *The Journal of Machine Learning Research*, 2014.

[8] Luo *et al.* Predicting student grade based on free-style comments using word2vec and ann by considering prediction results obtained in consecutive lessons. *EDM*, 2015.

[9] Meier *et al.* Personalized grade prediction: A data mining approach. In *ICDM*, 2015.

[10] Polyzou *et al.* Grade prediction with course and student specific models. In *PAKDD*. Springer, 2016.

[11] Reddy *et al.* Latent skill embedding for personalized lesson sequence recommendation. *arXiv preprint*, 2016.

[12] Romero *et al.* Data mining algorithms to classify students. In *EDM*, 2008.

[13] Sweeney *et al.* Next-term student performance prediction: A recommender systems approach. *arXiv preprint*, 2016.

Table 7: Models hyper-parameters.

Method	params	Fall 11	Spring 12	Fall 12	Spring 13	Fall 13	Spring 14	Fall 14	Spring 15
BiasOnly	α	1e-1	5e-2	5e-2	5e-2	1e-1	5e-2	1e-1	5e-2
	η	5e-4	5e-4	5e-4	5e-4	1e-3	1e-3	5e-3	5e-4
MF	α	5e-4	1e-4	5e-3	5e-3	5e-5	5e-5	5e-3	5e-3
	η	5e-4	5e-4	1e-3	1e-3	5e-4	5e-4	5e-3	5e-3
	dim	60	55	60	45	55	60	60	60
CSRM	α	1e-5	5e-4	5e-3	1e-3	5e-3	1e-5	1e-5	1e-5
	η	5e-5	1e-4	5e-4	5e-5	5e-5	5e-5	5e-5	5e-4
CKRMdep	α	5e-3	5e-3	1e-3	5e-3	1e-3	1e-3	5e-3	1e-3
	η	1e-2	5e-3	5e-3	5e-3	5e-4	1e-2	5e-3	5e-3
	λ	4e-1	4e-1	5e-1	2e-1	4e-1	4e-1	4e-1	3e-1
	dim	30	30	30	30	30	30	30	30
CKRMall	α	5e-5	5e-3	1e-3	1e-5	1e-3	1e-3	1e-4	5e-3
	η	5e-3	1e-2	5e-3	5e-3	1e-2	1e-2	1e-2	5e-3
	λ	5e-1	4e-1	5e-1	2e-1	5e-1	4e-1	5e-1	3e-1
	dim	30	20	30	30	30	30	20	30
CKRMtext	α	1e-4	1e-3	1e-3	1e-3	1e-3	1e-4	5e-5	1e-5
	η	1e-4	1e-4	5e-3	1e-4	1e-3	1e-4	1e-4	1e-4
	λ	4e-1	3e-1	5e-1	2e-1	4e-1	3e-1	5e-1	3e-1
	estimate-P	true	false	true	false	true	false	false	true

The headers of columns 3–10 denote the test term for each of the eight generated datasets. The parameter η denotes the learning rate for the SGD algorithm, the parameter "dim" denotes the number of latent dimensions used in the corresponding methods, and the parameter "estimate-P" for CKRMtext' models denotes whether the matrix $\mathbf{P}$ was estimated (true) or was used as indicator vectors (false).

[14] Thai-Nghe *et al.* Factorization models for forecasting student performance. In *EDM*, 2011.

[15] Thai-Nghe *et al.* Using factorization machines for student modeling. In *UMAP Workshops*, 2012.

Models Hyper-parameters

Table 7 shows the hyper-parameters of the selected models for each of the CKRM-based methods as well as the competing approaches for each of the eight subsets of the data that were generated (see Section 5.2).