Context-Aware Recommendation-Based Learning Analytics Using Tensor and Coupled Matrix Factorization

Faisal M. Almutairi, *Student Member, IEEE*, Nicholas D. Sidiropoulos, *Fellow, IEEE*, and George Karypis

***Abstract*—Student retention and timely graduation are enduring challenges in higher education. With the rapidly expanding collection and availability of learning data and related analytics, student performance can be accurately monitored, and possibly predicted ahead of time, thus, enabling early warning and degree planning "expert systems" to provide disciplined decision support to counselors, advisors, and educators. Previous work in educational data mining has explored matrix factorization techniques for grade prediction, albeit without taking contextual information into account. Temporal information should be informative as it distinguishes between the different class offerings and indirectly captures student experience as well. To exploit temporal and/or other kinds of context, we develop three approaches under the framework of collaborative filtering (CF). Two of the proposed approaches build upon coupled matrix factorization with a shared latent matrix factor. The third utilizes tensor factorization to model grades and their context, without introducing a new mode per context dimension as is common in the CF literature. The latent factors obtained can be used to predict grades *and* context, if desired. We evaluate these approaches on grade data obtained from the University of Minnesota. Experimental results show that fairly good prediction is possible even with simple approaches, but very accurate prediction is hard. The more advanced approaches can increase prediction accuracy, but only up to a point for the particular dataset considered.**

***Index Terms*—Alternating optimization, candecomp/parafac (CP) decomposition, collaborative filtering, coupled matrix factorization, matrix/tensor rank, predicting student performance, singular value decomposition (SVD), tensor factorization.**

I. Introduction

THERE has recently been growing interest in educational data mining [1] in general, and predicting student performance in particular [2]–[14]. The motivation behind this line of work is that student performance prediction can be leveraged to support instruction, advising, and counseling. As an example, predicting student performance in class activities during the semester [2], [15] can be used in early warning systems to identify students who are on the verge of failing a class in order to take a corrective action [4]. Other work has focused on predicting whether a student is able to perform a given task correctly [6]–[8], [12], [16], which can be used for class evaluation and exercise recommendation purposes. Moreover, leveraging course recommendation approaches as in [17]–[19], and methods for predicting final grades as in [3], [5], [10] can help to minimize the time-to-degree and build better academic planning tools. The work in this paper falls under the last category, as we aim to predict student performance at the course-level in terms of final grades in classes students have not yet taken. This can help in semester-to-semester course selection, recommendation of 'bridge' courses, and early warning systems. In this introduction, we provide a brief background focusing on the most relevant prior art and state the contribution of our work to education analytics and recommender systems in general.

A. Background and Related Work

Many researchers have proposed approaching the student performance prediction problem using *regression techniques* as in [20]. Recently, Elbadrawy, Studham, and Karypis [2] have presented collaborative multi-regression models which, unlike single regression-based approaches [4], allow for cross-student information sharing. They cross-leverage the advantages of regression-based models in accounting for students' interactions with Learning Management Systems (LMS) and factorization-based models in creating student-specific predictions [2]. Polyzou and Karypis [5] proposed models that utilize a judiciously chosen subset of the historical grade data when predicting grades for a specific course or a specific student: Course-Specific Regression (CSR), or Student-Specific Regression (SSR), respectively. In CSR, a grade is predicted by the student's prior grades with a linear regression model that determines how much each one of her/his past grades contributes. This regression vector is estimated by a model that utilizes only rows corresponding to students who took the course to be predicted in the students × courses grade matrix. As they pointed out, CSR uses the same regression vector for all students, which can be a limitation when applied on flexible academic programs where students have less common courses. To overcome this issue, in SSR, they eliminate courses that a student s has not taken

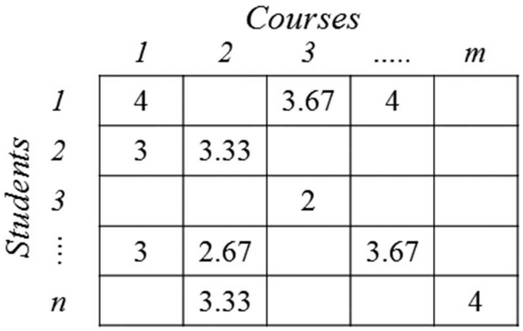
Manuscript received October 14, 2016; revised February 26, 2017 and April 24, 2017; accepted April 28, 2017. Date of publication May 18, 2017; date of current version July 18, 2017. This work was supported in part by NSF IIS-1447788. The guest editor coordinating the review of this paper and approving it for publication was Dr. Richard G. Baraniuk. *(Corresponding author: Nicholas D. Sidiropoulos.)*

F. M. Almutairi and N. D. Sidiropoulos are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: almut012@umn.edu; nikos@umn.edu).

G. Karypis is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: karypis@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2017.2705581



	Courses				
Students	1	2	3	.....	m
1	4		3.67	4	
2	3	3.33			
3			2		
⋮	3	2.67		3.67	
n		3.33			4

Fig. 1. Students × courses grade matrix sample.

from the grade matrix as well as students who have not taken the target course c, or do not have enough common courses with s to estimate a regression model that is personalized for each student [5].

Researchers have adapted *recommender system techniques* to the student performance prediction problem [3], [6], [8], [9], [11], and the course recommendation problem [17]–[19]. Typically, the users' ratings for items are represented in a users × items matrix [21]. Similarly, in the setting of latent factor and Matrix Factorization (MF) models, the students' grades are usually tabulated in a sparse students × courses matrix ,$\mathbf{G} \in \mathbb{R}^{n \times m}$, as in Fig. 1. In the context of grade prediction, various methods based on MF have been used to estimate the latent factors which produce representations for each student and course in order to be used to predict grades. The goal is to fill the missing values, which can be viewed as a matrix completion task [22]. The main idea here is to factor $\mathbf{G}$ as $\mathbf{G} \approx \mathbf{A}\mathbf{B}^T$, where $\mathbf{A} \in \mathbb{R}^{n \times F}$, $\mathbf{B} \in \mathbb{R}^{m \times F}$ and $F << \min(n, m)$ is the dimensionality of the latent space. Then missing entries in $\mathbf{G}$ are imputed based on $\mathbf{A}\mathbf{B}^T$, where the factors $\mathbf{A}$ and $\mathbf{B}$ are estimated from the available data by minimizing a suitable loss function. Sweeney, Lester, and Rangwala [3] explored estimating those factors for grade data via Singular Value Decomposition (SVD) and showed that grade prediction improves when SVD is followed by k-NN (k-Nearest Neighbor) post-processing, as detailed in [23]. It has been shown that adding global and local biases (for every student and course) to the MF model as in [5], [3] reduces the error in grade prediction.

The models we propose in this work draw from factor analysis for matrices and tensors, but can also be viewed under a recommender system 'lens'. It is therefore useful to provide the following classification for recommender systems, before we proceed to explain the specific modeling and optimization approaches proposed in this paper. Generally, models that are intended to predict the missing values (ratings or grades) are classified as follows [21]:

1) *Collaborative Filtering (CF):* in which predictions are calculated based on the historical ratings of all users collectively, either based on the similarities between users and items *(neighborhood-based)* or on latent factors and MF *(model-based)* [21].
2) *Content-based recommendations:* in which recommendation is provided depending on the similarity in the features of items that a user has rated or in the attributes of users who have rated the same items [24], [21].
3) *Hybrid between CF and content-based models:* various works have tried to cross-leverage the advantages of these two types of approaches. Our work here can be classified under this category.

The user and item features in CF models are learned from the data, which is assumed to exhibit a hidden low-dimensional structure [21]. On the other hand, in content-based recommendations, these features are given, e.g., a user's gender or a movie's genre. In the setting of student performance forecasting, the student's major or GPA, and the course's level or department are examples of such features.

In many situations, recommender system models that take additional *contextual information* into account provide more accurate recommendations as they are customized to each scenario [24]–[27]. Some of these methods are extending the traditional MF to incorporate useful side information besides historical ratings (grades). So-called context-aware recommender systems (CARS) can be categorized into three types: contextual *pre-filtering*, where the data is selected based on context; contextual *post-filtering*, where recommendations are filtered after they have been computed; and *contextual modeling*, where the context is accounted for while computing recommendations [28]. The proposed models in this paper fall under the last category, as context is exploited within the model. In our application domain of learning analytics, additional information such as time [7] has been used to enhance the accuracy of student performance prediction. González-Brenes [7] proposed a method where questions are clustered based on their latent representation derived from factoring the students × questions grade matrix. The resulting cluster centroids are interpreted as skills, and a model of skill acquisition is built for each student. This model is then used to predict a student's performance.

Coupled matrix factorization (CMF) has also been used in recommender systems where two or more matrices (one corresponding to ratings and the other(s) to side information) are jointly decomposed using one or more common latent factors corresponding to shared modes. Fang and Si [29] used a CMF method to incorporate implicit feedback in a system for online scientific community recommendation that also accounts for user content and/or item content. In the context of learning analytics, Lan *et al.* [13] (see also [12]) exploit context by jointly processing binary questions × students grade data together with questions × word-dictionary count data, using a sparse common matrix factor for the questions. In [13], the sum of the log-likelihood of the observed grades and the log-likelihood of the word counts given the latent factors is maximized. A Bernoulli model is used for the binary grades, and a Poisson model is used for the word counts. The context in both [29] and [13] is associated with only one dimension of the data – users (students) or items (questions). Recently, Sahebi *et al.* [14] proposed a CMF-based model that captures the improvement in student knowledge during multiple attempts at the same quiz, as part of the learning process.

Karatzoglou *et al.* [24] generalized the notion of MF representation in the context of CF by modeling data as a User × Item × Context1 × Context2 × $\cdots$ × Context(N-2) N-dimensional *tensor* where every type of context is introduced as a new mode

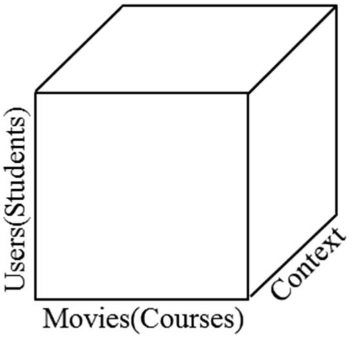


Fig. 2. Illustration of modeling context as a new mode.

in the tensor. For our particular application domain of predicting student performance, temporal side information has been exploited in [6] by modeling data as a three-way (student × task × time) tensor – see Fig. 2, where context would be time in this case. This is the prevalent approach for incorporating context in CF-based recommender systems [6], [24], [30]. Introducing a new mode for each context variable has two main drawbacks: first, the tensor size grows exponentially in the number of context variables; second, this usually yields an extremely sparse tensor. The reason is that each grade (rating) is usually given in only one context. Very sparse tensors require high rank to approximate (think, e.g., of a diagonal matrix, which is full rank if the elements on the diagonal are nonzero).

B. Contributions

1) Methodological: In this work, we propose three methods to incorporate additional information in the context of CF. In particular, we present two CMF models and one Low-Rank Tensor Factorization (LRTF) model. Intuitively, students sharing the same grade transcripts *and* course timestamps should be more predictive of each other than if they only share the former. If we decompose the grade and time (context) matrices using a common *student* latent factor, then two students will have the same latent representation if and only if they have the same grade transcripts *and* course timestamps, as illustrated in Section III-A1. In a similar manner, courses sharing the same student timestamps in addition to student grades are more predictive of each other than if they only share the same student grades. The reason is that grade statistics for a given course change over time, e.g., due to instructor variability, textbook changes, and broader trends such as grade inflation. If we decompose the grade and time matrices using a common *course* latent factor, then two courses will have the same latent factor if and only if they have the same grades *and* timestamps, as illustrated in Section III-A2. Using a common factor for students, and a common factor for courses simultaneously requires latent scaling and yields a LRTF model, as we will see in Section III-B.

Unlike [29] and [13], the side information that we exploit is associated with (student, course) pairs – it comes in the form of a matrix that has the same rows and columns as the grade matrix. We use i) iterative imputation, ii) student- and course-specific bias variables, and iii) a validation set to select the appropriate model rank. The approaches in [29] and [13] do not use iterative imputation, and employ rank regularization (via Frobenious norm penalties on the factor matrices) instead of explicit rank selection via validation. In [29], student/course-specific bias variables are not added to the models. We found that i)-iii) are important for obtaining competitive results in our application domain. Moreover, in contrast to the approach in [13], our models do not assume that the data follows any distribution.

In the case of LRTF, we model data in a tensor which has the grade matrix as the first frontal slab and the context matrix behind it. We factor this tensor using Candecomp/Parafac (CP) decomposition. If there are more than one type of context (context *dimensions*), we simply augment the tensor with more slabs at the back-end. Our modeling, in contrast to [6], [24], [30], maintains a common sparsity structure across the different matrices in CMF or tensor slabs in LRTF, facilitating low-rank modeling. Also, if the grade timestamp is modeled as a new mode and we wish to predict for next semester, we face the issue of predicting an entirely missing slab.[1] Our modeling of the context as a matrix in CMF models and as a tensor 'slab' in LRTF allows us to deal with this 'cold start' problem of an entire semester missing very nicely, as we in fact predict scattered entries instead of an entire slab.

Another advantage of our modeling approaches relative to [6], [24], [30] is that they predict not only the unseen grade or rating, but also the context in which the grade or rating will be earned/given. This is useful for forecasting course enrollments and other applications, such as targeted (context-sensitive) marketing and advertising in the case of movie recommendations.

2) Case Study: Although the models and methods that we propose can potentially be applied to other recommendation tasks (movies, products, restaurants, etc.), our original motivation comes from learning data analytics, and so we focus on student grade prediction. We apply our algorithms on real grade data from $\sim 10^4$ students and $\sim 10^3$ courses spanning 12 years from the College of Science and Engineering at the University of Minnesota. Experimental results show that fairly good prediction is possible even with simple approaches, but very accurate prediction is hard. The more advanced approaches can increase prediction accuracy, but only up to a point for the particular dataset considered. In particular, we verify that the proposed models and methods improve the baseline where no context is taken into account, and outperform other recent CF methods when predicting for a specific department and when predicting randomly missing grades.

Notation: Capital letters with underscore denote tensors, e.g. $\underline{\mathbf{X}}$; bold capital letters $\mathbf{A}, \mathbf{B}, \mathbf{C}$ denote matrices; bold small letters $\mathbf{a}, \mathbf{b}, \mathbf{c}$ denote column vectors; $\odot$ denotes Khatri-Rao (column-wise Kronecker) product; $\circledast$ denotes the Hadamard (element-wise) product; $\circ$ is the outer product; $\mathbf{A}^T$ denotes the transpose of $\mathbf{A}$; $\mathbf{A}^\dagger$ denotes the pseudo inverse of $\mathbf{A}$; $\mathbf{A}(i,:)$ or $\mathbf{a}_i$ denotes the ith row of $\mathbf{A}$ and $\mathbf{A}(:,j)$ denotes the jth column of

[1] An alternative is to use *relative* timestamps with respect to the time a student entered the program (reflecting student experience or 'seniority'). The drawback is that we risk clumping together students and courses that are temporally far apart, having very different learning experiences.

$\mathbf{A}$; $\underline{\mathbf{X}}(:,:,k)$ denotes the kth matrix slab of the three-way tensor $\underline{\mathbf{X}}$ taken perpendicular to the third mode; $\mathbf{D}_k(\mathbf{C}) := Diag(\mathbf{c}_k)$ is a diagonal matrix with the kth row of $\mathbf{C}$ on its diagonal; and $\text{vec}(\mathbf{A})$ is the vec-operator applied on a matrix $\mathbf{A}$ by stacking its columns into a vector.

II. Problem Formulation

Given a grade dataset indexed by (student, course) pairs, with contextual information, e.g., time, our main goal is to predict students' grades in courses they have not taken. In particular, we introduce three different models in Sections III-A and III-B in order to incorporate arbitrary side information alongside with historical grades to improve the accuracy of grade prediction. We denote the very sparse students × courses grade matrix $\mathbf{G}_o$ which is comprised by the observed grades for n students in m courses. Grades are encoded by mapping $[F, D, D+, C-, C, C+, B-, B, B+, A-, A]$ to numeric grades $[0, 1, 1.33, 1.67, 2, 2.33, 2.67, 3, 3.33, 3.67, 4]$, respectively. As for traditional MF techniques, $\mathbf{G}_o$ serves as the primary information source in our models [3].

In our experiments, we have tried two different types of side information: absolute time $\mathbf{T}_o$ in which the courses were taken, and student experience $\mathbf{E}_o$. Time in $\mathbf{T}_o$ is measured in semesters, while a student's experience in $\mathbf{E}_o$ is calculated by the number of semesters she/he has been in the program. The context is tabulated in a students × courses matrix, in the same way as the grade matrix $\mathbf{G}_o$ using (student, course) pairs. $\mathbf{T}_o(i,j)$/$\mathbf{E}_o(i,j)$ reveals the temporal/experience information corresponding to the grade $\mathbf{G}_o(i,j)$. For the remainder of this paper, we focus on the time context $\mathbf{T}_o$ in our formulation and results as we found that it is the most informative.

Each professor has her/his own way of grading even when they teach the same course. Moreover, year-to-year student cohort variation may cause a given professor to grade the same material differently through the years. By encoding every semester in $\mathbf{T}_o$ with a distinct digit, the various offerings of the same course can be distinguished. Specifically, semesters in $\mathbf{T}_o$ are mapped to consecutive integer numbers starting from 1. For instance, we encode Fall 2002 as 1, Spring 2003 as 2, Summer 2003 as 3 and so on – see Section IV-A. Although the context is formed in the same way, it is modeled and exploited in different ways by the three models, as we will see.

We train our models on the observed grades and their associated context after excluding a test set. After training each model, the task is to predict grades $\widehat{g}_{i,j}$ in the test data. Our models predict not only missing grades, but also the context in which the grade is earned, $\widehat{t}_{i,j}$.

III. Proposed Approaches

We present the three proposed methods and their algorithms in the following order: the two CMF models are explained in Section III-A and the LRTF model is presented in Section III-B.

A. Coupled Matrix Factorization (CMF)

Latent factor and traditional MF techniques explained in the introduction have been used in the context of recommender systems, specifically in CF-based methods [31]. Researchers have adapted recommendation techniques, e.g., based on MF to address the grade prediction problem [8], [9] interpreting students, courses, and grades as users, items, and ratings, respectively. Those models have shown very good results in the context of grade prediction using only historical grades of students [5], [31].

Contextual information, such as time (e.g., measured in semesters) and student seniority (experience) should be informative when predicting grades. For instance, students who have taken the same courses with similar grades are more predictive of each other if they actually took these classes in the same semesters (same time) than if they have a big time gap. A student might fail a class if taking it in her/his freshman year, but might get an A taking the same class in her/his senior year. This is an example of how student seniority can help as side information.

The main idea of CMF is to incorporate and exploit those pieces of contextual information into the traditional latent factor and MF models without over-complicating the solution. There are two CMF models proposed here, which are presented in detail in this section. Student and course bias terms are added to the formulation of these two models.

1) Coupled Matrix Factorization With Common Student Factors (CMFS): In this model, we decompose the students × courses grade and the students × courses context matrices using a common student latent factor. First, define a matrix $\mathbf{W} \in \mathbb{R}^{n \times m}$, such that:

$$\mathbf{W}(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{A}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where $\mathcal{A}$ includes the indices of observed grades. In the sequel, we also use $\overline{\mathbf{W}} \in \mathbb{R}^{n \times m}$ to denote the complement of $\mathbf{W}$. Hence, $\overline{\mathbf{W}}$ has ones at the indices of the missing entries of the grade matrix and zeros elsewhere. We can now define the 'complete' grade and time matrices in terms of the observed $\mathbf{G}_o$, $\mathbf{T}_o$ and missing $\mathbf{G}_m$, $\mathbf{T}_m$ grades and corresponding timestamps, respectively.

$$\begin{aligned} \mathbf{G} &:= \mathbf{W} \circledast \mathbf{G}_o + \overline{\mathbf{W}} \circledast \mathbf{G}_m, \\ \mathbf{T} &:= \mathbf{W} \circledast \mathbf{T}_o + \overline{\mathbf{W}} \circledast \mathbf{T}_m \end{aligned} \tag{2}$$

Then, CMFS can be formulated as follows:

$$\begin{aligned} \min_{\substack{\mathbf{G}_m, \mathbf{T}_m, \\ \mathbf{A}, \mathbf{B}_1, \mathbf{B}_2}} \; & \|\mathbf{W} \circledast \mathbf{G}_o + \overline{\mathbf{W}} \circledast \mathbf{G}_m - \mathbf{A}\mathbf{B}_1^T\|_F^2 \\ & + \|\mathbf{W} \circledast \mathbf{T}_o + \overline{\mathbf{W}} \circledast \mathbf{T}_m - \mathbf{A}\mathbf{B}_2^T\|_F^2 \end{aligned} \tag{3}$$

where $\mathbf{G}_o, \mathbf{T}_o \in \mathbb{R}^{n \times m}$ are the students × courses matrix of observed grades and the corresponding students × courses matrix of timestamps, respectively, $\mathbf{A} \in \mathbb{R}^{n \times F}$, $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{m \times F}$, and F is the dimensionality of latent space. Note that $\mathbf{G}_m, \mathbf{T}_m \in \mathbb{R}^{n \times m}$ are variables to be estimated. Also note that the entries of $\mathbf{G}_m$ corresponding to observed grades play no role, as they are zeroed out by the multiplication with $\overline{\mathbf{W}}$, and likewise for $\mathbf{T}_m$. These definitions are mainly used to facilitate concisely stating the problem and explaining the proposed algorithmic approaches. While the formulation in (3) has only one context, $\mathbf{T}_o$, we can add another context if desired, following the same concept of shared student latent factors.

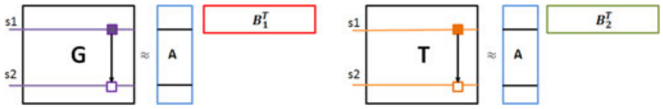


Fig. 3. Illustration of the basic idea behind CMFS.

To reformulate CMFS in a simpler form, define:

$$\mathbf{X} := \left[\, \mathbf{G} \;\vdots\; \mathbf{T} \,\right], \quad \mathbf{B}_c^T := \left[\, \mathbf{B}_1^T \;\vdots\; \mathbf{B}_2^T \,\right] \tag{4}$$

where $\mathbf{X} \in \mathbb{R}^{n \times 2m}$, and $\mathbf{B}_c \in \mathbb{R}^{2m \times F}$. Then, (3) can be reformulated as follows:

$$\min_{\mathbf{G}_m, \mathbf{T}_m, \mathbf{A}, \mathbf{B}_c} \|\mathbf{X} - \mathbf{A}\mathbf{B}_c^T\|_F^2 \tag{5}$$

Looking into equation (3), we can see that $\mathbf{A}$ and $\mathbf{B}_1$ are the low-rank factors of $\mathbf{G}$, and $\mathbf{A}$ and $\mathbf{B}_2$ are the low-rank factors of $\mathbf{T}$. Thus, the grades and the times share the same student latent factors matrix, $\mathbf{A}$. The underlying assumption of a student having the same latent representation in both domains (grade and time) is that there are few different types of students: say for example, ‘achievers’ that take courses early and do well; ‘strugglers’ that tend to delay taking advanced courses and have lower grades; ‘working’ students that may take courses at lower loads, etc. Therefore, a student type is associated with a grade pattern *and* a temporal pattern, and every student is a linear combination of these basic types.

In equation (5), note that $\mathbf{X}$ is an implicit function of $\mathbf{G}_m, \mathbf{T}_m$. One approach is to fix those, solve for $\mathbf{A}$ and $\mathbf{B}_c$ in (5) by Singular Value Decomposition (SVD) of $\mathbf{X}$, then fix $\mathbf{A}$ and $\mathbf{B}_c$ and impute the missing entries (i.e., update $\mathbf{G}_m, \mathbf{T}_m$), and continue to alternate between the two types of updates – see Section III-A4. If imputation is not desired, matrix completion approaches (e.g., Stochastic Gradient Descent (SGD)) can be used instead. However, we found in our experiments that imputation improves the prediction accuracy.

Intuitively, in order for two students to have the same latent factor representation (corresponding rows of $\mathbf{A}$), they must have the same grade transcripts *and* time profiles as illustrated in Fig. 3. The idea is that students sharing the same course timestamps in addition to grade transcripts are more predictive of each other than if they only have the same grades. For instance, if students s_1 and s_2 correspond to identical rows of $\mathbf{A}$ (horizontal black lines) in Fig. 3, we can predict the missing entry in s_2's grade in $\mathbf{G}$ based on s_1's grade.

2) Coupled Matrix Factorization With Common Course Factors (CMFC): In this model, the context matrix is exploited by jointly decomposing it with the students × courses grade matrix using a common course latent factor. CMFC is formulated as follows:

$$\min_{\substack{\mathbf{G}_m, \mathbf{T}_m, \\ \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}}} \|\mathbf{W} \circledast \mathbf{G}_o + \overline{\mathbf{W}} \circledast \mathbf{G}_m - \mathbf{A}_1\mathbf{B}^T\|_F^2 + \|\mathbf{W} \circledast \mathbf{T}_o + \overline{\mathbf{W}} \circledast \mathbf{T}_m - \mathbf{A}_2\mathbf{B}^T\|_F^2 \tag{6}$$

clearly, $\mathbf{A}_1$ and $\mathbf{B}$ are the low-rank factors of $\mathbf{G}$ (defined in (2)), and $\mathbf{A}_2$ and $\mathbf{B}$ are the low-rank factors of $\mathbf{T}$. Hence, the grade and the time matrices share the same course latent factors matrix, $\mathbf{B}$. In a similar manner to what we did to simplify CMFS, we define:

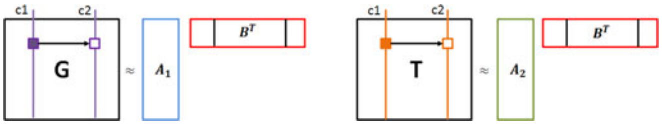


Fig. 4. Illustration of the basic idea behind CMFC.

$$\mathbf{Y} := \left[\, \mathbf{G}^T \;\vdots\; \mathbf{T}^T \,\right], \quad \mathbf{A}_c^T := \left[\, \mathbf{A}_1^T \;\vdots\; \mathbf{A}_2^T \,\right] \tag{7}$$

where $\mathbf{Y} \in \mathbb{R}^{m \times 2n}$, and $\mathbf{A}_c \in \mathbb{R}^{2n \times F}$. Then, (6) can be reformulated as follows:

$$\min_{\mathbf{G}_m, \mathbf{T}_m, \mathbf{A}_c, \mathbf{B}} \|\mathbf{Y} - \mathbf{B}\mathbf{A}_c^T\|_F^2 \tag{8}$$

problem (8) can be solved by alternating between SVD of $\mathbf{Y}$ and imputation for missing entries by updating $\mathbf{G}_m$ and $\mathbf{T}_m$.

Studies have shown that grade statistics for a given course change over time, e.g., due to instructor variability, textbook changes, and broader trends such as grade inflation. In CMFC formulation, in order for two courses to have the same latent factor representation (corresponding columns of $\mathbf{B}^T$), they must have the same student grades *and* time profiles as illustrated in Fig. 4. In other words, two courses will be similar if they have been co-taken by students in near-by semesters in addition to having similar grade distribution. This formulation will pick up the change of grades over time. The idea behind this model is that courses sharing the same student timestamps in addition to grades are more predictive of each other than if they only have the same student grades. For example, if two courses, c_1 and c_2, correspond to identical columns of $\mathbf{B}^T$ (vertical black lines) in Fig. 4, we can predict the missing entry in c_2 grades in $\mathbf{G}$ based on the corresponding c_1 grade.

3) Student and Course Biases for CMF Models: For more accurate prediction of grades, we add student and course bias terms to the grade and context matrices. This is inspired by the improvement that user and item biases bring to movie recommendation systems [21], [32]. Modeling how a student is likely to perform on average (student bias), and how difficult a course is on average (course bias) have also been shown to be effective in grade prediction using different but related models and approaches [2], [5], [6]. After incorporating biases, the formulation of CMFS becomes:

$$\min_{\substack{\mathbf{G}_m, \mathbf{T}_m, \\ \mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \\ \mathbf{b}_s, \mathbf{b}_c, \mathbf{t}_s, \mathbf{t}_c}} \|\mathbf{W} \circledast \mathbf{G}_o + \overline{\mathbf{W}} \circledast \mathbf{G}_m - \mathbf{A}\mathbf{B}_1^T - \mathbf{b}_s\mathbf{1}^T - \mathbf{1}\mathbf{b}_c^T\|_F^2 + \|\mathbf{W} \circledast \mathbf{T}_o + \overline{\mathbf{W}} \circledast \mathbf{T}_m - \mathbf{A}\mathbf{B}_2^T - \mathbf{t}_s\mathbf{1}^T - \mathbf{1}\mathbf{t}_c^T\|_F^2 \tag{9}$$

Where $\mathbf{b}_s \in \mathbb{R}^n$ is the student grade bias vector, $\mathbf{b}_c \in \mathbb{R}^m$ is the course grade bias vector, $\mathbf{t}_s \in \mathbb{R}^n$ is the student context (time) bias vector, and $\mathbf{t}_c \in \mathbb{R}^m$ is the course context bias vector.

Similarly for CMFC:

$$\min_{\substack{\mathbf{G}_m,\mathbf{T}_m,\\ \mathbf{A}_1,\mathbf{A}_2,\mathbf{B},\\ \mathbf{b}_s,\mathbf{b}_c,\mathbf{t}_s,\mathbf{t}_c}} \|\mathbf{W} \circledast \mathbf{G}_o + \overline{\mathbf{W}} \circledast \mathbf{G}_m - \mathbf{A}_1\mathbf{B}^T - \mathbf{b}_s\mathbf{1}^T - \mathbf{1}\mathbf{b}_c^T\|_F^2 + \|\mathbf{W} \circledast \mathbf{T}_o + \overline{\mathbf{W}} \circledast \mathbf{T}_m - \mathbf{A}_2\mathbf{B}^T - \mathbf{t}_s\mathbf{1}^T - \mathbf{1}\mathbf{t}_c^T\|_F^2 \tag{10}$$

After we train the two CMF models on the grade training set and its corresponding context, the prediction of the grade that a student i is going to obtain in a course j and the prediction of the time in which this grade will be earned are given in (11) for CMFS model:

$$\begin{aligned}\widehat{g}_{i,j} &= \mathbf{b}_s(i) + \mathbf{b}_c(j) + \mathbf{A}(i,:)\mathbf{B}_1^T(:,j),\\ \widehat{t}_{i,j} &= \mathbf{t}_s(i) + \mathbf{t}_c(j) + \mathbf{A}(i,:)\mathbf{B}_2^T(:,j)\end{aligned} \tag{11}$$

Similarly for CMFC, grades and context are predicted using (12):

$$\begin{aligned}\widehat{g}_{i,j} &= \mathbf{b}_s(i) + \mathbf{b}_c(j) + \mathbf{A}_1(i,:)\mathbf{B}^T(:,j),\\ \widehat{t}_{i,j} &= \mathbf{t}_s(i) + \mathbf{t}_c(j) + \mathbf{A}_2(i,:)\mathbf{B}^T(:,j)\end{aligned} \tag{12}$$

4) CMF Algorithm: Since CMFS and CMFC can be solved using the same algorithm (up to transposition), we focus on the CMFS formulation in (9).

Recall $\mathbf{G}$ and $\mathbf{T}$ as defined in equation (2). Then, define:

$$\mathbf{G}_b := \mathbf{G} - \mathbf{b}_s\mathbf{1}^T - \mathbf{1}\mathbf{b}_c^T, \quad \mathbf{T}_b := \mathbf{T} - \mathbf{t}_s\mathbf{1}^T - \mathbf{1}\mathbf{t}_c^T \tag{13}$$

$$\mathbf{X}_b := \left[\mathbf{G}_b \;\vdots\; \mathbf{T}_b\right]. \tag{14}$$

The algorithm for CMFS is as follows.

Algorithm 1: CMFS formulation (9).

```
    Input: $\mathbf{G}_o$, $\mathbf{T}_o$ and $\nu$
1:  Scaling: scale the context matrix $\mathbf{T}_o$ with $\nu$ – important
    for accurate prediction.
2:  Initialization: impute missing entries in $\mathbf{G}_m$ with the
    average of the observed grades in $\mathbf{G}_o$; same for $\mathbf{T}_m$;
    $\mathbf{b}_s=\mathbf{b}_c=\mathbf{t}_s=\mathbf{t}_c=\mathbf{0}$
3:  Repeat
      • Update $\mathbf{G}_b$ and $\mathbf{T}_b$ using (13)
      • $\mathbf{A}, \mathbf{B}_c$ (defined in (4)) $\leftarrow$ SVD($\mathbf{X}_b$)
      • Update grade bias vectors $\mathbf{b}_s = \frac{(\mathbf{G}-\mathbf{A}\mathbf{B}_1^T-\mathbf{1}\mathbf{b}_c^T)\mathbf{1}}{m}$, and
        $\mathbf{b}_c = \frac{(\mathbf{G}-\mathbf{A}\mathbf{B}_1^T-\mathbf{b}_s\mathbf{1}^T)^T\mathbf{1}}{n}$
      • Update context bias vectors $\mathbf{t}_s = \frac{(\mathbf{G}-\mathbf{A}\mathbf{B}_2^T-\mathbf{1}\mathbf{t}_c^T)\mathbf{1}}{m}$, and
        $\mathbf{t}_c = \frac{(\mathbf{G}-\mathbf{A}\mathbf{B}_2^T-\mathbf{t}_s\mathbf{1}^T)^T\mathbf{1}}{n}$
      • Impute missing values of $\mathbf{G}$ by updating
        $\mathbf{G}_m = \mathbf{A}\mathbf{B}_1^T + \mathbf{b}_s\mathbf{1}^T + \mathbf{1}\mathbf{b}_c^T$
      • Impute missing values of $\mathbf{T}$ by updating
        $\mathbf{T}_m = \mathbf{A}\mathbf{B}_2^T + \mathbf{t}_s\mathbf{1}^T + \mathbf{1}\mathbf{t}_c^T$
    until convergence (the normalized difference of the cost
    at two successive iterations $< \epsilon$).
    Return: $\mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \mathbf{b}_s, \mathbf{b}_c, \mathbf{t}_s$, and $\mathbf{t}_c$
```

B. Low-Rank Tensor Factorization (LRTF)

Considerable work has been done in recent years towards incorporating context with the goal of building more personalized recommender systems, for which context turns out playing an important role. The context-aware CF model based on *Tensor Factorization* introduced in [24] models data as a User × Item × Context tensor. A similar approach was used in [6] to predict student performance, modeling data as a three-mode tensor (student × task × time). Although this technique improves the prediction accuracy by exploiting the context, it increases the sparsity of the data by introducing context as a new mode, therefore increasing the rank required for accurate approximation, especially without imputation. The reason is that a very sparse tensor with a random sparsity pattern requires rank in the order of the number of nonzero elements, as a rank-one tensor is spent explaining each available element, and there is very slow error 'roll-off' as one increases the model's rank.

We propose instead the LRTF model, where a context matrix (e.g., time $\mathbf{T}_o$) is introduced as another slab behind the main historical grades matrix $\mathbf{G}_o$. The rest of this section is structured as follows. In section III-B1, we review the Candecomp/Parafac (low-rank) decomposition for tensors, our LRTF formulation is explained in Section III-B2, and an algorithm to solve this formulation is presented in Section III-B3.

1) Candecomp/Parafac (CP) Decomposition: We summarize the basics of CP decomposition as it is essential in our LRTF formulation. Decomposing a three-way tensor as a sum of outer products (rank-one three-way tensors) as a data analysis technique was proposed independently by Carroll and Chang [33] (they called it Candecomp) and Harshman [34] (who called it Parafac). The CP decomposes a three-way array $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ into a sum of F rank-one tensors [35], i.e.,

$$\underline{\mathbf{X}} \approx \sum_{f=1}^{F} \mathbf{a}_f \circ \mathbf{b}_f \circ \mathbf{c}_f \tag{15}$$

where F is a positive integer, $\mathbf{a}_f \in \mathbb{R}^I$, $\mathbf{b}_f \in \mathbb{R}^J$, and $\mathbf{c}_f \in \mathbb{R}^K$ [36]. A CP solution is usually expressed in terms of the factor matrices $\mathbf{A} \in \mathbb{R}^{I \times F}$, $\mathbf{B} \in \mathbb{R}^{J \times F}$, and $\mathbf{C} \in \mathbb{R}^{K \times F}$, which have the vectors $\mathbf{a}_f$, $\mathbf{b}_f$ and $\mathbf{c}_f$ as columns, respectively, i.e., $\mathbf{A} = [\mathbf{a}_1 \;\; \mathbf{a}_2 \;\; \ldots \;\; \mathbf{a}_F]$ and likewise for $\mathbf{B}$ and $\mathbf{C}$. Let $\underline{\mathbf{X}}(:,:,k)$ denote the kth slice (frontal 'slab') of $\underline{\mathbf{X}}$. Then (15) can be written as:

$$\underline{\mathbf{X}}(:,:,k) \approx \mathbf{A}\mathbf{D}_k(\mathbf{C})\mathbf{B}^T \tag{16}$$

where $\mathbf{D}_k(\mathbf{C})$ is a diagonal matrix holding the kth row of $\mathbf{C}$ on its diagonal [35].

2) LRTF Model Formulation: As a first step, we model the grade matrix $\mathbf{G}$ which includes the observed and missing grades as defined in (2) and its context $\mathbf{T}$ as a tensor with two frontal slices – $\mathbf{G}$ in front, and the context matrix we wish to use behind it[2], i.e.,

$$\underline{\mathbf{X}}(:,:,1) = \mathbf{G}, \quad \underline{\mathbf{X}}(:,:,2) = \mathbf{T} \tag{17}$$

[2] If we desire to exploit more than one contextual information, a third slice of context can be added, etc.

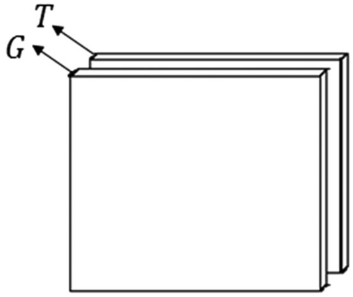


Fig. 5. Illustration of modeling grade data and its context in LRTF.

Therefore $\underline{\mathbf{X}} \in \mathbb{R}^{n\times m\times 2}$ in the case where only one context is added behind $\mathbf{G}$, where the first mode describes n students, the second mode describes m courses, and the third mode describes the number of contexts added to the grade matrix. Modeling data in frontal slices is illustrated in Fig. 5. The advantage of this modeling is that adding a context does not increase sparsity, it maintains exactly the same sparsity pattern as in $\mathbf{G}_o$. In the LRTF model, we typically use an alternating optimization algorithm to estimate the factor matrices $\mathbf{A} \in \mathbb{R}^{n\times F}$, $\mathbf{B} \in \mathbb{R}^{m\times F}$ and $\mathbf{C} \in \mathbb{R}^{2\times F}$ of the CP decomposition of $\underline{\mathbf{X}}$. After every update of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ we use them to impute for missing grades and context. Overall, LRTF can be formulated as follows:

$$\min_{\substack{\mathbf{G}_m,\mathbf{T}_m,\\ \mathbf{A},\mathbf{B},\mathbf{C}}} \quad \|\mathbf{W} \circledast \mathbf{G}_o + \overline{\mathbf{W}} \circledast \mathbf{G}_m - \mathbf{A}\mathbf{D}_1(\mathbf{C})\mathbf{B}^T\|_F^2 + \|\mathbf{W} \circledast \mathbf{T}_o + \overline{\mathbf{W}} \circledast \mathbf{T}_m - \mathbf{A}\mathbf{D}_2(\mathbf{C})\mathbf{B}^T\|_F^2 \qquad (18)$$

Clearly, the grade and context matrices share the same $\mathbf{A}$ and $\mathbf{B}$ factors in the above LRTF formulation. Therefore, in order for two students who take the same classes to be more predictive of each other, they must share similar time profiles (or any other context) as well. Solving problem (18) requires relatively small rank due to the imputation that occurs within every iteration (the update of $\mathbf{G}_m$ and $\mathbf{T}_m$) and the fact that sparsity is not affected by adding context slabs.

Similar to CMFS, and CMFC, in our experience, adding student and course biases always improves the accuracy of grade prediction for LRTF as well. After accounting for the student and course bias vectors for the grade matrix and its context, (18) becomes:

$$\min_{\substack{\mathbf{G}_m,\mathbf{T}_m,\mathbf{A},\mathbf{B},\mathbf{C}\\ \mathbf{b}_s,\mathbf{b}_c,\mathbf{t}_s,\mathbf{t}_c}} \quad \|\mathbf{W} \circledast \mathbf{G}_o + \overline{\mathbf{W}} \circledast \mathbf{G}_m - \mathbf{A}\mathbf{D}_1(\mathbf{C})\mathbf{B}^T - \mathbf{b}_s\mathbf{1}^T - \mathbf{1}\mathbf{b}_c^T\|_F^2 + \|\mathbf{W} \circledast \mathbf{T}_o + \overline{\mathbf{W}} \circledast \mathbf{T}_m - \mathbf{A}\mathbf{D}_2(\mathbf{C})\mathbf{B}^T - \mathbf{t}_s\mathbf{1}^T - \mathbf{1}\mathbf{t}_c^T\|_F^2 \qquad (19)$$

3) LRTF Algorithm: We now present the algorithm that solves the LRTF formulation to estimate the CP factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, which are subsequently used to predict the grades and their context as follows:

$$\widehat{g}_{i,j} = \mathbf{b}_s(i) + \mathbf{b}_c(j) + \mathbf{A}(i,:)\mathbf{D}_1(\mathbf{C})\mathbf{B}^T(:,j),$$
$$\widehat{t}_{i,j} = \mathbf{t}_s(i) + \mathbf{t}_c(j) + \mathbf{A}(i,:)\mathbf{D}_2(\mathbf{C})\mathbf{B}^T(:,j) \qquad (20)$$

Algorithm 2: LRTF formulation (19).

```
   Input: $\mathbf{G}_o$, $\mathbf{T}_o$ and $\nu$
Scaling: scale the context matrix $\mathbf{T}_o$ with $\nu$ – important
   for accurate prediction.
Initialization: impute missing entries in $\mathbf{G}_m$ with the
   average of the observed grades in $\mathbf{G}_o$; same for $\mathbf{T}_m$;
   $\mathbf{b}_s$=$\mathbf{b}_c$=$\mathbf{t}_s$=$\mathbf{t}_c$=$\mathbf{0}$; Initialize for $\mathbf{A}, \mathbf{B}, \mathbf{C}$ using N-way
   Toolbox [37] to provide a better initialization as it uses
   algebraic methods (eigen-decomposition) for
   initialization.
Repeat
   • Update $\mathbf{G}_b$ and $\mathbf{T}_b$ using (13)
   • Update $\mathbf{A} \leftarrow \mathbf{X}_b([\mathbf{D}_1(\mathbf{C})\mathbf{B}^T, \mathbf{D}_2(\mathbf{C})\mathbf{B}^T])^\dagger$
   • Update $\mathbf{B} \leftarrow \mathbf{Y}_b([\mathbf{D}_1(\mathbf{C})\mathbf{A}^T, \mathbf{D}_2(\mathbf{C})\mathbf{A}^T])^\dagger$
   • Update $\mathbf{c}_1 \leftarrow \text{vec}(\mathbf{G}_b)(\mathbf{B} \odot \mathbf{A})^\dagger$
   • Update $\mathbf{c}_2 \leftarrow \text{vec}(\mathbf{T}_b)(\mathbf{B} \odot \mathbf{A})^\dagger$
   • Update $\mathbf{b}_s = \frac{(\mathbf{G}-\mathbf{A}\mathbf{D}_1(\mathbf{C})\mathbf{B}^T-\mathbf{1}\mathbf{b}_c^T)\mathbf{1}}{m}$, and
     $\mathbf{b}_c = \frac{(\mathbf{G}-\mathbf{A}\mathbf{D}_1(\mathbf{C})\mathbf{B}^T-\mathbf{b}_s\mathbf{1}^T)^T\mathbf{1}}{n}$
   • Update $\mathbf{t}_s = \frac{(\mathbf{G}-\mathbf{A}\mathbf{D}_2(\mathbf{C})\mathbf{B}^T-\mathbf{1}\mathbf{t}_c^T)\mathbf{1}}{m}$, and
     $\mathbf{t}_c = \frac{(\mathbf{G}-\mathbf{A}\mathbf{D}_2(\mathbf{C})\mathbf{B}^T-\mathbf{t}_s\mathbf{1}^T)^T\mathbf{1}}{n}$
   • Impute missing values of $\mathbf{G}$ by updating
     $\mathbf{G}_m = \mathbf{A}\mathbf{D}_1(\mathbf{C})\mathbf{B}^T + \mathbf{b}_s\mathbf{1}^T + \mathbf{1}\mathbf{b}_c^T$
   • Impute missing values of $\mathbf{T}$ by updating
     $\mathbf{T}_m = \mathbf{A}\mathbf{D}_2(\mathbf{C})\mathbf{B}^T + \mathbf{t}_s\mathbf{1}^T + \mathbf{1}\mathbf{t}_c^T$
until convergence (the normalized difference of the cost
at two successive iterations < $\epsilon$)
Return: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{b}_s$, $\mathbf{b}_c$, $\mathbf{t}_s$, and $\mathbf{t}_c$
```

Recall $\mathbf{X}_b$ defined in (14) and define:

$$\mathbf{Y}_b := \left[\, \mathbf{G}_b^T \;\vdots\; \mathbf{T}_b^T \,\right]. \qquad (21)$$

The algorithmic steps to solve the LRTF model are summarized in Algorithm 2.

IV. Experimental Design

In this section we provide necessary information about the setup and the design of experiments we performed to test the three proposed models, CMFS, CMFC, and LRTF, on real educational data. We describe the features of this grade dataset and explain the construction of the context we used as a side information in Section IV-A. Our baseline where no context is added and other methods used for comparison are summarized in Section IV-B. In Section IV-C, we explain the different test sets used for testing alongside with the metrics used for evaluation. Finally, in the last Section IV-D, we clarify our model selection strategy used to choose parameters for all methods.

A. Dataset and Context

The experimental results were obtained using real historical grade data from the College of Science and Engineering (CSE) students at the University of Minnesota. Experimental results are shown for two different datasets. Dataset 1 contains all the grades of students of CSE for any course they have

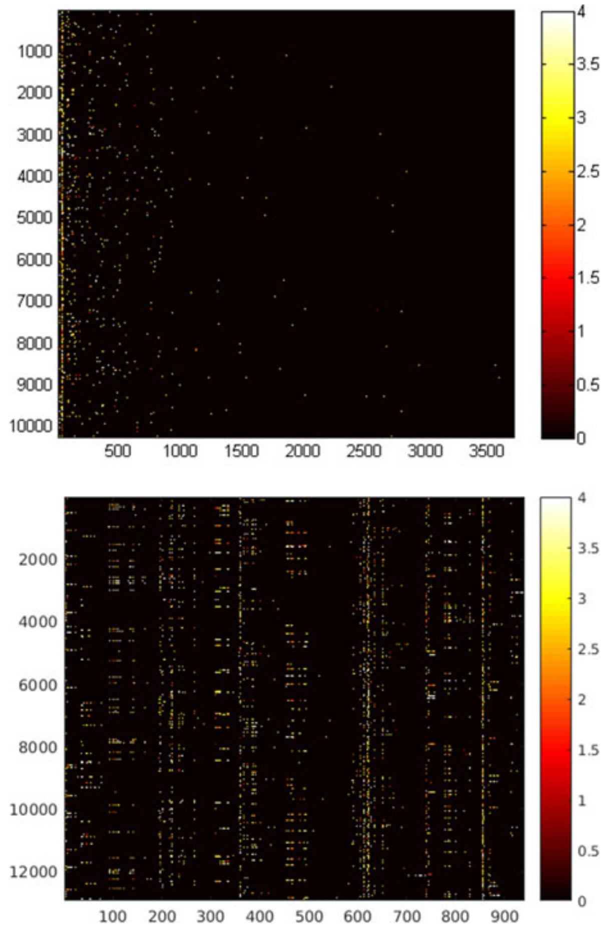


Fig. 6. Visualization of the grade matrix $\mathbf{G}_o$ obtained from Dataset 1 (top) and Dataset 2 (bottom).

TABLE I
Description of Features of Grade Datasets

Feature	Dataset 1	Dataset 2
# of students (n)	10,245	12,938
# of courses (m)	3712	941
# of observations	244,086	269,073
sparsity	99.36%	97.79%
period	Fall 2002 - Fall 2013	Fall 2002 - Spring 2015

taken between Fall 2002 until Fall 2013, including courses offered by other colleges. Dataset 2 includes grades of students of CSE strictly for courses under one of the following departments: Aerospace Engineering, Biomedical Engineering, Chemical Engineering, Chemistry, Civil Engineering, Computer science, Electrical Engineering, Material Science, Mathematics, Mechanical Engineering, Physics, and Statistics. The $\mathbf{G}_o$ matrices for the two datasets are visualized in Fig. 6 through which we can view their sparsity. Clearly, Dataset 2 is less sparse as courses are limited to CSE departments. Table I summarizes the two datasets and their features.

We also apply our methods to subsets of data corresponding to specific departments, where students are more highly correlated. Thus, we extracted the Electrical and Computer Engineering (ECE) and Chemistry (Chem) departments from Dataset 2. The subsets of data corresponding to students who declared these two majors are summarized in Table II.

TABLE II
Description of Features of Grade Datasets for ECE and Chem Departments (Subsets of Dataset 2)

Feature	ECE	Chem
# of students (n)	1,306	1,106
# of courses (m)	702	632
# of observations	27,171	22,960
sparsity	97.04%	96.72%

For each observed dataset, we computed CMFS, CMFC, and LRTF models with two different types of contextual information: absolute time $\mathbf{T}_o$ indicating the semester in which a grade was earned; and student experience $\mathbf{E}_o$, reflecting seniority when taking a course. Throughout our experiments, we used these two contexts individually and together with each one of the proposed models. We found out that using the time context individually is the most informative as it provides the best grade prediction. We think that adding the experience context $\mathbf{E}_o$ besides the time $\mathbf{T}_o$ did not improve the prediction due to the correlation between these two context – student experience can be inferred from (is implicit in) the absolute time information. Therefor, we show results in the next section for simulations conducted using the time context $\mathbf{T}_o$ alongside with observed grades $\mathbf{G}_o$. As explained in Section II, we map semesters to consecutive integer numbers starting from 1. Hence, the maximum value in $\mathbf{T}_o$ is 34 for Dataset 1 and 38 for Dataset 2. An illustrative example of this mapping is shown in Fig. 7.

B. Comparison With Other Methods

We compare the performance of CMFS, CMFC and LRTF with their baseline where no context is used, as well as the following methods:

1) *Baseline :* Factoring the grade matrix $\mathbf{G}$ as defined in (2) with iterative imputation and grade bias terms in the same way as the proposed models, but without including any context. The baseline is formulated as follows:

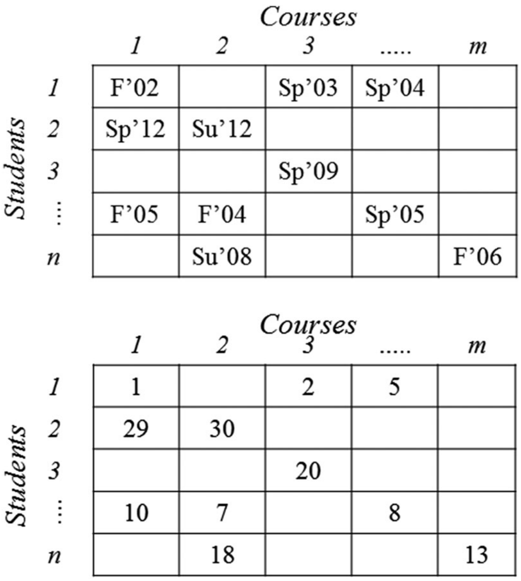
$$\min_{\substack{\mathbf{G}_m,\mathbf{A},\\ \mathbf{B},\mathbf{b}_s,\mathbf{b}_c}} \|\mathbf{W} \circledast \mathbf{G}_o + \overline{\mathbf{W}} \circledast \mathbf{G}_m - \mathbf{A}\mathbf{B}^T - \mathbf{b}_s \mathbf{1}^T - \mathbf{1}\mathbf{b}_c^T\|_F^2 \tag{22}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the low-rank factors of $\mathbf{G}$.

2) *Matrix Factorization (MF):* The MF approach is described in [5]. To match the MF model with our notation, we rewrite its formulation as follows:

$$\min_{\substack{\mathbf{A},\mathbf{B}\\ \mu,\mathbf{b}_s,\mathbf{b}_c,}} \sum_{g_{i,j}\in\mathbf{G}_o} (g_{i,j} - \mu - \mathbf{b}_s(i) - \mathbf{b}_c(j) - \mathbf{A}(i,:)\mathbf{B}^T(:,j))^2$$
$$+ \lambda(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{b}_s\|_2^2 + \|\mathbf{b}_c\|_2^2) \tag{23}$$

where μ is a global bias, $\mathbf{b}_s \in \mathbb{R}^n$ and $\mathbf{b}_c \in \mathbb{R}^m$ are the student and course bias vectors, respectively, and $\mathbf{A} \in \mathbb{R}^{n\times F}$ and $\mathbf{B} \in \mathbb{R}^{m\times F}$ are the low-rank latent factors.



	1	2	3	.....	m
1	F'02		Sp'03	Sp'04	
2	Sp'12	Su'12			
3			Sp'09		
⋮	F'05	F'04		Sp'05	
n		Su'08			F'06

	1	2	3	.....	m
1	1		2	5	
2	29	30			
3			20		
⋮	10	7		8	
n		18			13

Fig. 7. Students × courses semester matrix (top) and an illustrative example of the mapping done to construct $\mathbf{T}_o$ (bottom).

3) BiasOnly: As described in [5], only global and local bias terms are considered in equation (23). Biases are estimated using the MF formulation above with $F = 0$.

4) Context Tensorization (CXT): First, grade data is modeled in a student × course × time (semester) tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{n \times m \times t}$ similar to [6], where t is the number of semesters. Then, we estimate the factors of the CP decomposition for this tensor. The formulation of CXT is as follows:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \quad \sum_{k=1}^{t} \|\underline{\mathbf{W}}(:,:,k) \circledast (\underline{\mathbf{Y}}(:,:,k) - \mathbf{A}\mathbf{D}_k(\mathbf{C})\mathbf{B}^T)\|_F^2 + \lambda(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2 + \|\mathbf{C}\|_F^2) \tag{24}$$

where $\underline{\mathbf{W}}(i,j,k) = 1$ if $\underline{\mathbf{Y}}(i,j,k)$ is available, 0 otherwise. This formulation is the prevailing one in context-aware CF-based recommender systems. We estimate the CP low rank factors $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ using column-wise alternating optimization (CCD++) [38], which allows us to easily handle missing data. We also tried using SGD for this purpose [38], but with generally inferior results.

C. Test Sets and Evaluation Metrics

To assess the proposed models, we test their prediction of grades in a test set, $\mathbf{G}_{\text{test}}$, which is either: 1) the last semester or 2) randomly selected 10% of the observed grades in $\mathbf{G}_o$. We should mention that when we test on the last semester for Dataset 1, we discard Fall 2013 and Summer 2013 and test on Spring 2013, as Summer 2013 and Fall 2013 do not have enough observed grades (Dataset 1 was recorded before the end of the Fall 2013 semester). The cardinality (N) of the test set for theses two cases for Dataset 1 and Dataset 2, and the number of predicted grades for ECE and Chem departments for Spring 2015 are shown in Table III. We denote the time context associated with $\mathbf{G}_{\text{test}}$ as $\mathbf{T}_{\text{test}}$. Grades in $\mathbf{G}_{\text{test}}$ and their corresponding context $\mathbf{T}_{\text{test}}$ are excluded from $\mathbf{G}_o$ and $\mathbf{T}_o$, respectively.

TABLE III
CARDINALITY (N) OF THE VARIOUS TEST SETS, $|\mathbf{G}_{\text{test}}|$

	Dataset 1	Dataset 2	ECE	Chem
last semester	14,723	9,176	872	824
random 10%	24,407	26,908	–	–

The grade prediction accuracy is evaluated using the *Root Mean Squared Error* (RMSE) and *Mean Absolute Error* (MAE) of $\mathbf{G}_{\text{test}}$ as defined in (25) and (26), respectively. Same metrics are used to calculate the accuracy of predicting context in $\mathbf{T}_{\text{test}}$.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{g_{i,j} \in \mathbf{G}_{\text{test}}} (g_{i,j} - \widehat{g_{i,j}})^2} \tag{25}$$

$$\text{MAE} = \frac{\sum_{g_{i,j} \in \mathbf{G}_{\text{test}}} |g_{i,j} - \widehat{g_{i,j}}|}{N} \tag{26}$$

When we predict for the last semester, $\mathbf{G}_{\text{test}}$ and $\mathbf{T}_{\text{test}}$ are fixed, and therefore RMSE and MAE are directly used for evaluation. However, for the case of testing on a randomly selected 10% of the grades, we run each experiment 3 times and present the Average RMSE (AvgRMSE) and the Average MAE (AvgMAE).

D. Model Selection and Training

Parameters in CMFS, CMFC, LRTF, and the methods in Section IV-B are selected based on the performance of these models on a validation set $\mathbf{G}_{\text{val}}$, which is randomly selected 10% of observed grade data $\mathbf{G}_o$ not including the test set $\mathbf{G}_{\text{test}}$.

For all models (except for BiasOnly), we perform a greedy search on the best model rank (R) – note that $R = F + 2$ in the models that have two bias terms, where F is the number of columns of $\mathbf{A}$ (or $\mathbf{A}_1$). For our models, the baseline, and MF, we let R range from 3 to 30 with increments of 1. For CXT model, we perform a search on R in the range from 1 to 100 with increments of 1.

Similarly, we search for the best value of ν used to scale the context matrix in the range of 0 to 5 with an increment of 0.05; note that $\nu = 0$ yields the baseline model. The best ν varies depending on the type of test set $\mathbf{G}_{\text{test}}$ we are testing on as described in Section IV-C. We found that $\nu = 1/2$ gives the best prediction when $\mathbf{G}_{\text{test}}$ is the last semester, while $\nu = 1/10$ gives the best prediction in the case of testing on $\mathbf{G}_{\text{test}}$ as randomly selected 10% of observed data. In MF, BiasOnly, and CXT methods, we search for the best λ (the regularization parameter) from 0 to 16 in increments of 0.05. For both cases of $\mathbf{G}_{\text{test}}$, the best prediction is found when $\lambda = 0.65$ for MF and $\lambda = 0.25$ for BiasOnly. When testing on subsets of the data, in MF $\lambda = 0.8$ and $\lambda = 0.7$ give the best prediction for ECE and Chem, respectively. While for BiasOnly $\lambda = 0.3$ gives the best prediction for both departments.

While the cost of our model is monotonically improving with iterations, the prediction RMSE/MAE is not. We found that the RMSE is a convex (U-shaped) function of the number of iterations when we solve for the proposed models. The stopping criterion used to terminate model fitting iterations is based on the cost function as explained in algorithms 1 and 2. We monitor the prediction RMSE on $\mathbf{G}_{\text{val}}$ and terminate when it starts rising again.

Once we tune parameters based on $\mathbf{G}_{\text{val}}$, we train our models using the provided algorithmic procedures on the grade matrix $\mathbf{G}_o$ and its context $\mathbf{T}_o$. Then, grades in the test set $\mathbf{G}_{\text{test}}$ and their context $\mathbf{T}_{\text{test}}$ are predicted using the last model fitted before we terminated the iterations by equations (11), (12) and (20) for CMFS, CMFC, and LRTF, respectively. The prediction error metrics are then calculated using the formulas provided in Section IV-C.

V. Experimental Results

In this section we show the performance of CMFS, CMFC, and LRTF models which exploit the time context $\mathbf{T}_o$ alongside with $\mathbf{G}_o$ and compare them to the baseline, MF, BiasOnly, and CXT – the first three approaches use only the observed grade matrix $\mathbf{G}_o$, while CXT uses the semester information as third mode, as described in Section IV-B. The performance is measured in terms of the accuracy of predicting grades in the test set $\mathbf{G}_{\text{test}}$. For our models and the baseline, we show results for the best three model ranks for each dataset, and present the results of MF and CXT with their best rank. We also show how well our methods predict the time context in $\mathbf{T}_{\text{test}}$. As we test on two different test sets, last semester and randomly missing grades, we dedicate one section for each test set type.

TABLE IV
Prediction Error of the Proposed vs. Existing Methods Measured by RMSE and MAE for Grades in $\mathbf{G}_{\text{test}}$ (Last Semester) With Their Best Rank(s), R

	Dataset 1			**Dataset 2**		
Method	R	RMSE	MAE	R	RMSE	MAE
Baseline	3	**0.6659**	0.4728	3	**0.6628**	0.4778
	4	0.6706	0.4784	4	0.6709	0.4880
	5	0.6884	0.4964	5	0.6732	0.4919
MF	3	0.6702	0.4667	3	0.6638	0.4737
BiasOnly	2	0.6824	0.4800	2	0.6712	0.4843
CMFS	3	0.6818	0.4865	3	**0.6672**	0.4831
	4	0.6877	0.4904	4	0.6694	0.4858
	5	0.6925	0.4972	5	0.6701	0.4859
CMFC	3	**0.6716**	0.4747	3	0.6674	0.4816
	4	0.6781	0.4855	4	0.6699	0.4863
	5	0.6879	0.4945	5	0.6756	0.4919
LRTF	9	0.6797	0.4820	10	0.6680	0.4809
	10	0.6780	0.4816	11	0.6689	0.4816
	11	0.6778	0.4825	12	0.6698	0.4830

TABLE V
Prediction Error of the Proposed Methods Measured by RMSE and MAE for Timestamps in $\mathbf{T}_{\text{test}}$ (Last Semester) With Their Best Ranks, R

	Dataset 1			**Dataset 2**		
Method	R	RMSE	MAE	R	RMSE	MAE
CMFS	3	1.7599	1.2662	3	**1.3329**	0.8273
	4	2.1210	1.3918	4	2.0686	1.4342
	5	2.2021	1.4660	5	2.1386	1.4946
CMFC	3	**1.6435**	1.1009	3	1.3374	0.8335
	4	2.1820	1.4429	4	2.0975	1.4699
	5	2.3025	1.5728	5	2.2731	1.6980
LRTF	9	2.4888	1.6791	10	2.1249	1.2882
	10	2.5199	1.7098	11	2.1698	1.3402
	11	2.6359	1.7910	12	2.2195	1.3857

A. Prediction of Last Semester

The performance results shown in this section are calculated on the last semester test set. Table IV shows the prediction error of students' final grades evaluated by RMSE and MAE for our models and the methods in comparison. The CXT model is not included in this comparison, as the entire last semester slab $\underline{\mathbf{Y}}(:,:,t)$ is missing from the training data, and this is something that CXT cannot handle. Even if we perfectly model the other slabs, we have no data point for the last semester, and thus cannot even scale the model to the last semester. This is a well-known drawback of context tensorization approaches, known as 'cold start'. Amongst our models, CMFC with rank $R = 3$ works best for Dataset 1, while CMFS with $R = 3$ gives the smallest error for Dataset 2. The results on the last semester were unexpected, as our methods with their best rank outperform BiasOnly but they do not improve the baseline or MF with $R = 3$. This might be due to the nature of the dataset we are using (the best models have very low rank, and student and course BiasOnly predicts almost as well as any other method, which indicates either very simple or very challenging 'extreme' data); or the fact that we resort to alternating optimization as we cannot fit the models to optimality. Another explanation for these results is that we tune model parameters for randomly missing validation set and test on the last semester test set which have different natures. To resolve this, we used the semester before the last one to select models and it did not work well, as different courses are usually offered in the fall and spring semesters. It is worth mentioning here that imputing for missing grades while fitting the model helps, as this is the main difference between our baseline and MF – note the results of baseline vs. MF in Table IV. For each dataset, we highlight the smallest RMSE produced by our models and the baseline to make it easier for the reader to compare.

In Table V, we show the prediction error of our models for the time at which grades in the last semester test set were obtained. Note that the correct time values in $\mathbf{T}_{\text{test}}$ which we are predicting are the same which correspond to last semester. Recall that we scale $\mathbf{T}_o$ by $\nu = 1/2$, hence every semester adds 0.5 including summers. CMFS with $R = 3$ can predict time with less than three semesters error – roughly within one year error as we account for summer semesters. We should mention that

TABLE VI
PREDICTION ERROR OF THE PROPOSED VS. EXISTING METHODS MEASURED BY RMSE AND MAE FOR GRADES IN $\mathbf{G}_{\text{test}}$ (LAST SEMESTER FOR ECE AND CHEM DEPARTMENTS)

	ECE			**Chem**		
Method	R	RMSE	MAE	R	RMSE	MAE
Baseline	3	**0.6534**	0.4754	3	**0.6898**	0.4939
	4	0.6673	0.5028	4	0.7060	0.5169
	5	0.6614	0.4854	5	0.6989	0.5114
MF	3	0.6735	0.4834	3	0.7074	0.5077
BiasOnly	2	0.6564	0.4828	2	0.7063	0.5161
CMFS	3	0.6569	0.4803	3	0.6927	0.5027
	4	0.6700	0.5004	4	0.7201	0.5275
	5	0.6722	0.5043	5	0.7209	0.5283
CMFC	4	0.6518	0.4815	4	0.6892	0.5043
	5	0.6481	0.4762	5	**0.6741**	0.4934
	6	**0.6412**	0.4702	6	0.6851	0.5029
LRTF	6	0.6566	0.4761	6	0.6899	0.4985
	7	0.6579	0.4793	7	0.6911	0.4995
	10	0.6568	0.4785	8	0.6942	0.5019

TABLE VII
PREDICTION ERROR OF THE PROPOSED VS. EXISTING METHODS MEASURED BY THE AVERAGE ERROR (AVGRMSE AND AVGMAE) OF THREE DIFFERENT HELD-OUT GRADE TEST SETS $\mathbf{G}_{\text{test}}$ (RANDOMLY MISSING) WITH THEIR BEST RANK(S), R

	Dataset 1			**Dataset 2**		
Method	R	AvgRMSE	AvgMAE	R	AvgRMSE	AvgMAE
Baseline	3	0.6132	0.4448	3	**0.5830**	0.4345
	4	0.6123	0.4421	4	0.5856	0.4367
	5	0.6196	0.4466	5	0.5853	0.4350
MF	3	**0.6118**	0.4416	3	0.5862	0.4411
BiasOnly	2	0.6258	0.4598	2	0.5861	0.4414
CXT (CCD++)	2	0.6983	0.4955	5	0.6143	0.4620
CXT (SGD)	5	0.7444	0.5230	5	0.6112	0.4614
CMFS	3	0.6159	0.4469	3	0.5827	0.4345
	5	0.6118	0.4419	4	0.5816	0.4339
	6	0.6192	0.4478	6	0.5843	0.4353
CMFC	3	**0.6091**	0.4409	3	0.5805	0.4322
	4	0.6116	0.4413	4	0.5799	0.4320
	5	0.6168	0.4447	5	0.5828	0.4339
LRTF	5	0.6127	0.4433	8	**0.5790**	0.4314
	6	0.6114	0.4420	9	0.5797	0.4321
	7	0.6121	0.4424	10	0.5808	0.4328

although the smallest error in Table V is RMSE $= 1.3329$, this is not the best time prediction that can be obtained. The reason is that we tune the models based on the prediction of grades as it is our main interest and then predict for the context.

t4 *Predicting for a single Department:* One might be curious about how the different methods compare when focusing on a single department where students are highly correlated. We show the results of the prediction for students of ECE and Chem (subsets of Dataset 2) in Table VI. The CMFC model improves the prediction of the baseline, MF, and BiasOnly models in both cases – the improvement is highlighted.

TABLE VIII
PREDICTION ERROR OF THE PROPOSED METHODS MEASURED BY THE AVERAGE ERROR (AVGRMSE AND AVGMAE) OF THREE DIFFERENT HELD-OUT TIMESTAMP TEST SETS $\mathbf{T}_{\text{test}}$ (RANDOMLY MISSING) WITH THEIR BEST RANKS, R

	Dataset 1			**Dataset 2**		
Method	R	AvgRMSE	AvgMAE	R	AvgRMSE	AvgMAE
CMFS	3	**0.2158**	0.1498	3	**0.1952**	0.1299
	5	0.2421	0.1637	4	0.2058	0.1377
	6	0.2610	0.1778	6	0.2370	0.1621
CMFC	3	0.2247	0.1615	3	0.2062	0.1427
	4	0.2416	0.1722	4	0.2142	0.1471
	5	0.2502	0.1741	5	0.2220	0.1505
LRTF	5	0.2371	0.1634	8	0.2224	0.1471
	6	0.2432	0.1699	9	0.2286	0.1505
	7	0.2502	0.1701	10	0.2318	0.1522

TABLE IX
ESTIMATES OF THE STANDARD DEVIATION (SD) OF THE PREDICTION RMSE AND MAE FOR THE PROPOSED MODELS USING 100 DIFFERENT RANDOMLY HELD-OUT $\mathbf{G}_{\text{test}}$ AND $\mathbf{T}_{\text{test}}$ AND THE LISTED RANKS, R

		$\mathbf{G}_{\text{test}}$		$\mathbf{T}_{\text{test}}$	
Method	R	SD(RMSE)	SD(MAE)	SD(RMSE)	SD(MAE)
CMFS	6	3.2E-03	2.0E-03	2.6E-03	1.5E-03
CMFC	5	3.6E-03	2.3E-03	3.3E-03	2.1E-03
LRTF	8	3.2E-03	2.2E-03	2.7E-03	1.5E-03

B. *Prediction of Randomly Missing 10% of Data*

In this section we show the prediction error of testing on randomly missing 10% of data. Table VII shows the prediction error for the proposed models, baseline, MF, BiasOnly, and CXT. For this test set, our models outperform MF, BiasOnly, and CXT methods and improve the baseline prediction. CMFC gives the smallest error among our models for Dataset 1, while LRTF provides the best prediction for Dataset 2. The smallest AvgRMSE provided by the proposed models and the best of the other methods are highlighted to point out the improvement. We get stable results - the RMSEs for the different held-out subsets are close to each other. For example, the three RMSE values are 0.5787, 0.5798, and 0.5785 for the LRTF model for Dataset 2 with $R = 8$ – other results are qualitatively similar.

For this test set, our models predict the time with smaller error than in the case of last semester, as is clear from Table VIII. We scale $\mathbf{T}_o$ by $\nu = 1/10$ in this case, hence every semester adds 0.1, including summers. For Dataset 2, the time was predicted using CMFS with less than two semesters error, AvgRMSE $= 0.1952$.

To give an idea about statistical significance of the results, we report estimates of the standard deviation (SD) of the prediction RMSE and MAE for the proposed models using 100 different randomly held-out test sets. In Table IX, SD estimates are shown for Dataset 2 for the listed ranks.

VI. Conclusion

In this work, we extended the traditional MF method in the context of CF to incorporate contextual information in a different way than is commonly done in the recommender system literature. Utilizing *matrix factorization* and *tensor factorization* we proposed models that allow for flexible integration of side information, *without amplifying sparsity / increasing the needed model rank*. They also can handle the 'cold start' problem when predicting for a certain context, such as next semester. These models share the feature of factoring the grade matrix and the context with a common factor to exploit the context in which a grade was earned. Algorithms to solve the presented formulations were provided.

These models were tested on actual grade data obtained from the University of Minnesota with time context measured in semesters. We conducted a careful comparison with our baseline and existing methods that are used in the setting of predicting student performance. Comparisons were made on the prediction of grades in two types of test set, last semester and randomly missing grades. Although the results were a bit disappointing for last semester predictions as our modeling did not improve the baseline or MF, they showed a promising prediction improvement for the case of testing on a single department. In the case of predicting randomly missing grades, the proposed models outperform earlier methods and the smallest error is provided by CMFC for Dataset 1 and LRTF for Dataset 2.

Another aspect that the presented modeling could be used for is context prediction, e.g., course enrollment forecasting. For the randomly missing grade prediction, the time was predicted using CMFS with less than two semesters error on average, $\text{AvgRMSE} = 0.1952$.

Acknowledgment

The authors would like to thank S. Smith for his help with the CXT simulations.

References

[1] R. S. Baker and K. Yacef, "The state of educational data mining in 2009: A review and future visions," *JEDM-J. Educ. Data Min.*, vol. 1, no. 1, pp. 3–17, 2009.

[2] A. Elbadrawy, R. Studham, and G. Karypis, "Collaborative multi-regression models for predicting students' performance in course activities," in *Proc. 5th Int. Conf. Learn. Analytics Knowl.*, Poughkeepsie, NY, USA, Mar 2015, pp. 103–107.

[3] M. Sweeney, J. Lester, and H. Rangwala, "Next-term student grade prediction," in *IEEE Int. Conf. Big Data*. Santa Clara, CA, USA: IEEE, Nov 2015, pp. 970–975.

[4] R. Barber and M. Sharkey, "Course correction: Using analytics to predict course success," in *Proc. 2nd Int. Conf. Learn. Analytics Knowl.* Vancouver, BC, Canada: ACM, Apr. 2012, pp. 259–262.

[5] A. Polyzou and G. Karypis, "Grade prediction with course and student specific models," in *Proc. 20th Pacific-Asia Conf. Adv. Knowl. Discovery Data Min.*, Auckland, New Zealand: Springer, Apr. 2016, pp. 89–101.

[6] N. Thai-Nghe, T. Horváth, and L. Schmidt-Thieme, "Factorization models for forecasting student performance," in *Proc. 4th Int. Conf. Educ. Data Mining.* Eindhoven, The Netherlands, Jul. 2011, pp. 11–20.

[7] J. González-Brenes, "Modeling skill acquisition over time with sequence and topic modeling," in *Proc. 18th Int. Conf. Artif. Intell. Stat.*, San Diego, CA, USA, May 2015, pp. 296–305.

[8] A. Toscher and M. Jahrer, "Collaborative filtering applied to educational data mining," in *Proc. KDD Cup Workshop*, Washington, DC, USA, Jul. 2010, pp.17–28.

[9] N. Thai-Nghe, A. L. Drumond Krohn-Grimberghe, and L. Schmidt-Thieme, "Recommender system for predicting student performance," in *Proc. 1st Workshop Recommender Syst. Technol. Enhanc. Learn.*, Barcelona, Spain, Sep. 2010, pp. 2811–2819.

[10] M. Al-Barrak and M. Al-Razgan, "Predicting students final GPA using decision trees: A case study," *Int. J. Inf. Educ. Technol.*, vol. 6, no. 7, pp. 528–533, 2016.

[11] H. Bydžovská, "Are collaborative filtering methods suitable for student performance prediction?" in *Prog. Artifi. Intell.: 17th Portuguese Conf. Artif. Intell.*, Coimbra, Portugal, Sep. 2015, pp. 425–430.

[12] A. S. Lan, A. E. Waters, C. Studer, and R. G. Baraniuk, "Sparse factor analysis for learning and content analytics," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1959–2008, 2014.

[13] A. S. Lan, C. Studer, A. E. Waters, and R. G. Baraniuk, "Joint topic modeling and factor analysis of textual information and graded response data," in *Proc. 6th Int. Conf. Edu. Data Mining*, Memphis, TN, USA, Jul. 2013, pp. 324–325.

[14] S. Sahebi, Y. Lin, and P. Brusilovsky, "Tensor factorization for student modeling and performance prediction in unstructured domain," in *Proc. 9th Int. Conf. Educ. Data Mining*. NC, USA: IEDMS, Jun./Jul. 2016, pp. 502–506.

[15] K. E. Arnold and M. D. Pistilli, "Course signals at Purdue: Using learning analytics to increase student success," in *Proc. 2nd Int. Conf. Learn. Analytics Knowl*, Vancouver, BC, Canada: ACM, 2012, pp. 267–270.

[16] N. Thai-Nghe, L. Drumond, T. Horváth, and L. Schmidt-Thieme, "Using factorization machines for student modeling," in *Workshop Poster Proc. 20th Conf. User Modeling, Adaptation, Personalization–UMAP 2012*, Montreal, QC, Canada. Jul. 2012.

[17] T. Denley, "Course recommendation system and method," US Patent App. 13/441 063, Jan. 2013.

[18] S. Ray and A. Sharma, "A collaborative filtering based approach for recommending elective courses," in *Int. Conf. Inf. Intell., Syst., Technol. Manage.*, Gurgaon, India: Springer, Mar. 2011, pp. 330–339.

[19] A. Elbadrawy and G. Karypis, "Domain-aware grade prediction and top-n course recommendation," in *Proc. 10th ACM Recommender Syst. Conf.* Boston, MA, USA: ACM, Sep. 2016, pp. 183–190.

[20] H. Cen, K. R. Koedinger, and B. Junker, "Learning factors analysis–A general method for cognitive model evaluation and improvement," in *Int. Conf. Intell. Tutoring Syst.*, Jhongli, Taiwan: Springer, Jun. 2006, pp. 164–175.

[21] P. Melville and V. Sindhwani, "Recommender systems," in *Encyclopedia of Machine Learning*. New York, NY, USA: Springer-Verlag, 2011, pp. 829–838.

[22] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[23] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDD Cup Workshop*, San Jose, CA, USA, Aug. 2007, pp. 5–8.

[24] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, Barcelona, Spain: ACM, Sep. 2010, pp. 79–86.

[25] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Trans. Info. Syst.*, vol. 23, no. 1, pp. 103–145, 2005.

[26] K. Oku, S. Nakajima, J. Miyazaki, and S. Uemura, "Context-aware SVM for context-dependent information recommendation," in *Proc. 7th Int. Conf. Mobile Data Manage.*, Nara, Japan: IEEE, May 2006, p. 109.

[27] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proc. 5th ACM Conf. Recommender Syst.*, Chicago, IL, USA: ACM, Oct. 2011, pp. 301–304.

[28] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. New York, NY, USA: Springer-Verlag, 2010.

[29] Y. Fang and L. Si, "Matrix co-factorization for recommendation with rich side information and implicit feedback," in *Proc. 2nd Int. Workshop Inf. Heterogeneity Fusion Recommender Syst.*, Chicago, IL, USA: ACM, 2011, pp. 65–69.

[30] L. Xiong, X. Chen, T. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. 2010 SIAM Int. Conf. Data Mining*. Columbus, OH, USA: SIAM, Apr. 2010, pp. 211–222.

[31] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*. New York, NY, USA: Springer-Verlag, 2010.

[32] R. M. Bell, Y. Koren, and C. Volinsky, "The BellKor 2008 Solution to the Netflix Prize, year = 2008," Tech. Rep. [Online]. Available: http://www.netflixprize.com/assets/ProgressPrize2008_BellKor.pdf

[33] J. D. Carroll and J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[34] R. Harshman, "Foundations of the Parafac procedure: Models and conditions for an explanatory multimodal factor analysis," *UCLA Working Papers in Phonetics 16*, pp. 1–84, 1970.

[35] A. Stegeman and N. D. Sidiropoulos, "On Kruskals uniqueness condition for the Candecomp/Parafac decomposition," *Linear Algebra Appl.*, vol. 420, no. 2, pp. 540–552, 2007.

[36] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[37] C. A. Andersson and R. Bro, "The n-way toolbox for matlab," *Chemometr. Intell. Lab. Syst.*, vol. 52, no. 1, pp. 1–4, 2000.

[38] S. Smith, J. Park, and G. Karypis, "An exploration of optimization algorithms for high performance tensor completion," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Salt Lake City, UT, USA, Nov. 2016, pp. 359–371.

Faisal M. Almutairi received the B.Sc degree with first-class honor in electrical engineering—electronics and communications from Qassim University (QU), Saudi Arabia, in 2012, the M.S. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 2016, where he is working toward the Ph.D. degree in the Department of Electrical and Computer Engineering. He has served as a Teaching Assistant at QU for multiple theory and laboratory courses. His research interests include signal processing, data analytics, and optimization.

Nicholas D. Sidiropoulos (F'09) received the Diploma degree in electrical engineering from the Aristotelian University of Thessaloniki, Thessaloniki, Greece, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 1988, 1990 and 1992, respectively. He served as an Assistant Professor at the University of Virginia, Associate Professor at the University of Minnesota, and Professor at TU Crete, Greece. Since 2011, he has been with the University of Minnesota, Minneapolis, MN, USA, where he currently holds an ADC Chair in digital technology. His research interests include topics in signal processing theory and algorithms, optimization, communications, and factor analysis—with a long-term interest in tensor decomposition and its applications. His current focus is primarily on signal and tensor analytics for learning from big data. He received the NSF/CAREER award in 1998, and the IEEE Signal Processing (SP) Society Best Paper Award in 2001, 2007, and 2011. He received the 2010 IEEE SP Society Meritorious Service Award, and the 2013 Distinguished Alumni Award from the Department of ECE, University of Maryland. He served as the IEEE SP Society Distinguished Lecturer (2008–2009), and as Chair of the IEEE Signal Processing for Communications and Networking Technical Committee (2007–2008). He is a Fellow of EURASIP (2014).

George Karypis is a Distinguished McKnight University Professor and an ADC Chair of Digital Technology at the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, MN, USA. His research interests include the areas of data mining, high-performance computing, information retrieval, collaborative filtering, bioinformatics, cheminformatics, and scientific computing. His research has resulted in the development of software libraries for serial and parallel graph partitioning (METIS and ParMETIS), hypergraph partitioning (hMETIS), for parallel Cholesky factorization (PSPASES), for collaborative filtering-based recommendation algorithms (SUGGEST), clustering high dimensional datasets (CLUTO), finding frequent patterns in diverse datasets (PAFI), and for protein secondary structure prediction (YASSPP). He has coauthored more than 280 papers on these topics and two books ("Introduction to Protein Structure Prediction: Methods and Algorithms" (Wiley, 2010) and "Introduction to Parallel Computing" (Publ. Addison Wesley, 2003, 2nd edition)). In addition, he is serving on the program committees of many conferences and workshops on these topics, and on the editorial boards of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Knowledge Discovery from Data*, *Data Mining and Knowledge Discovery*, *Social Network Analysis and Data Mining Journal*, *International Journal of Data Mining and Bioinformatics*, *Journal on Current Proteomics*, *Advances in Bioinformatics*, and *Biomedicine and Biotechnology*.