Iterative Discriminant Tensor Factorization for Behavior Comparison in Massive Open Online Courses

Xidao Wen
University of Pittsburgh
Pittsburgh, PA
xidao.wen@pitt.edu

Yu-Ru Lin*
University of Pittsburgh
Pittsburgh, PA
yurulin@pitt.edu

Xi Liu
Texas A&M University
College Station, TX
xiliu.tamu@gmail.com

Peter Brusilovsky
University of Pittsburgh
Pittsburgh, PA
peterb@pitt.edu

Jordan Barría Pineda
University of Pittsburgh
Pittsburgh, PA
jab464@pitt.edu

ABSTRACT

The increasing utilization of massive open online courses has significantly expanded global access to formal education. Despite the technology's promising future, student interaction on MOOCs is still a relatively under-explored and poorly understood topic. This work proposes a multi-level pattern discovery through hierarchical discriminative tensor factorization. We formulate the problem as a hierarchical discriminant subspace learning problem, where the goal is to discover the shared and discriminative patterns with a hierarchical structure. The discovered patterns enable a more effective exploration of the contrasting behaviors of two performance groups. We conduct extensive experiments on several real-world MOOC datasets to demonstrate the effectiveness of our proposed approach. Our study advances the current predictive modeling in MOOCs by providing more interpretable behavioral patterns and linking their relationships with the performance outcome.

ACM Reference Format:
Xidao Wen, Yu-Ru Lin, Xi Liu, Peter Brusilovsky, and Jordan Barría Pineda. 2019. Iterative Discriminant Tensor Factorization for Behavior Comparison in Massive Open Online Courses. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3308558.3313713

1 INTRODUCTION

While massive open online courses (MOOCs) have been attracting an ever-increasing number of students, the low completion rate (between 5%-10% [19]) has been a major obstacle to the transformative potentials of MOOCs. The predictive analysis of student performance thus emerged an an important research topic offering insights to platform developers and instructors in arranging proper learning support and allocating resources to students. To find informative predictors, researchers have focused on extracting features from students' interaction with the MOOC platform, such as watching videos, working on assignments, and viewing or contributing to discussion forums. Applied predictive models range from standard machine learning methods [23, 28] to more advanced ones such as deep learning [7]. These prediction models could be useful in predicting learning outcomes but are notably limited in helping understand the underlying learning behavior.

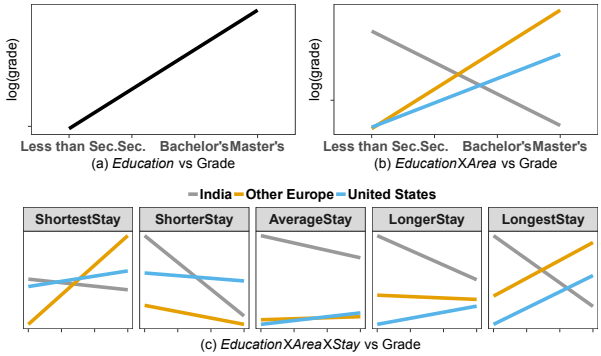

Figure 1: Association analysis of student performance on MOOCs. (a) shows the positive association between the grade and education level; (b) shows the mixed associations when including the area where the student is from; (c) further breaks down the observed groups into different level of activity (five quantiles w.r.t. the number of days students remained active on the platform).

There is abundant work that aims to better understand the behavior patterns that relate to the learning outcomes. For example, Coleman et al. [4] correlates each "behavior topic" to the learning outcomes based on topic modeling. However, the research to date fails to consider the multi-dimensional nature of the features and their potential interactions in outcome learning. Meanwhile, the famous Simpson's paradox points out that the direction of an association at the population-level may be reversed within the subgroups comprising that population [20]. To further explain this in the context of the MOOC platform, we use the Edx MOOC dataset [14] and investigate the factors associated with the students' grade on it.

We extract the education, area (the region where the student is from), the number of active days and the final grade of each student in the course "Introduction to Computer Science and Programming" offered by MITx in Spring 2013. The course had more than 44,000 online participants. Figure 1 shows the association between the selected factors and the final grades of the students. By comparing

*Corresponding author.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313713

Figure 1(a) and Figure 1(b), we observe that when the factor of area is considered, mixed associations occur. For example, the subgroup of Indian participants exhibited a negative association between education level and grade, which potentially suggests a more conservative understanding of the relationship between educational background and course outcome. Moreover, when we consider the number of active days, as in Figure 1(c), we notice that for the participants from the "other European" area, the positive association has a strong presence— but only with *ShortestStay* and *LongestStay*. Thus, in comparison to typical correlation analysis, a prediction model that can take advantage of the multi-way interactions of the features could potentially yield better performance.

A growing body of research seeks to resolve Simpson's paradox through causality inference [34]. However, causality is not the focus of this study as we aim for a data-driven approach to subgroup comparisons and explorations. In many cases, this can be interesting and important even in non-causal settings. A straightforward solution is to perform regression with feature interactions, or use Factorization Machines [37] that allow for the estimation of high-order interaction effects. However, the drawback of these methods is that they offer little understanding of the underlying multi-way learning behavior dynamics and their relationship to the learning outcomes. On the other hand, Factorization models like Matrix Factorization (2-way) and Tensor Factorization (m-way) are able to provide an in-depth understanding of meaningful behavior dynamics [9], but there are a few drawbacks preventing them from being more widely adopted by researchers in the field. First, the associations are isolated, with each of them capturing a certain trend of the behaviors separately (e.g., Figure 2(c)). Second, conventional pattern discovery through factorization models provides little support for contrasting pattern exploration that aims to identify the shared and discriminative behavior characteristics among different groups of users. Being able to do this can tremendously improve knowledge of user behaviors in the context of user group analysis.

In this work, we formulate the problem of understanding learning behavior in MOOCs as (1) the simultaneous factorization of the association between students' multi-aspect features and their performance, and (2) the iterative discovery of interpretable shared and discriminative patterns at multiple levels. The critical challenge is how to utilize the multi-way interaction of the features while providing interpretable patterns to help domain experts understand the learning dynamics. We propose a tensor-based learning method—iterative Discriminative tensor factorization (`iDisc`)—that discovers the common and discriminative learning patterns at multiple levels, and based on which we project users to a latent space (i.e. *embedding* for the downstream prediction tasks) to identify the association between the multi-way interaction of the features and the students' performance. To this end, we first represent the behaviors of the students from the opposite performance groups as *coupled tensors*. Since the coarse-grained joint factorization of these behavior tensors may not be capable of revealing behavior patterns at the subgroup level, `iDisc` iteratively performs discriminative pattern discovery at multiple levels. To increase the interpretability of the entire pattern space, we also introduce the inference of pattern hierarchy. To make the solution capable of handling unseen students, we project the students' behavior tensors into a latent space, by considering the multi-way interactions at different levels as the loading matrix. The empirical studies with the dataset from different MOOC platforms have shown the promising results on the effectiveness and efficiency of `iDisc`.

Contributions Our contributions can be summarized as follows:

- We formulate the problem of identifying the multi-way feature interaction with interpretable pattern discovery for understanding user behavior on the MOOC platforms.
- We propose a framework of iterative discriminant factorization for multi-way data. By factorizing the residual tensors at each level, our method enables the discovery of common and discriminative patterns at different granular levels. To ensure the parsimony of the discovered structure, we employ sparse learning to effectively capture enforcing relationships between the top-level and bottom-level patterns.
- We perform extensive experimentation of our methodology using several real-world datasets, and show the efficiency and interpretability of our proposed method.

2 RELATED WORK

Predictive Modeling in MOOCs. There are several types of predictive models in MOOCs that are closely related to this work. One direction is to utilize more complex feature types, including higher-order n-gram representations of learner activity data. For instance, features are constructed using the occurrence of pre-defined sequential activities [44], or from sequential pattern mining [11, 17, 26, 40]. Another line of work proposes to utilize the temporal nature of the activity data for student success prediction. Qiu et al. [36] propose a latent dynamic factor graph (LadFG) to model and predict learning behavior in MOOCs. LadFG captures the dynamic information and homophily correlations between students. It also projects students' learning behavior into a latent continuous space for predicting student performance. Another approach is the latent variable modeling as a way of inferring complex relationships between predictors [13, 25, 27]. For instance, Halawa et al. [13] explore the use of count-based learning activity features to predict dropout; this approach suggests that both observable learner activity and dropout are driven by latent, unobservable "persistence" factors.

Common and Discriminative Subspace Learning. The increasing availability of data from diverse sources has enabled the study on joint analysis of heterogeneous data. Gupta et al. [12] propose a joint Non-negative Matrix Factorization (NMF) approach to separate the common and discriminative subspace. Following the same idea, Kim et al. [22] relax the framework by requiring the shared subspace to be *similar* rather than strictly identical. Wen et al. [46, 47] propose a data-driven method of jointly factorizing the paired tensors with auxiliary tensors that preserve the common and distinctive signals. However, existing works focus primarily on pattern discoveries. This limits their use in downstream tasks (e.g., prediction or classification).

Multi-Level Tensor Factorization. Multi-Level Tensor Factorization addresses the problem of approximating the hierarchical low-rank tensor format. This process allows the representation of the tensors in a nested subspace, in one of Tree-Tucker format [30], tensor train format [29], or tensor networks format [3]. Huang et al. [18] employ a tree-guided learning via tensor decomposition

Table 1: Description of Notations.

Symbol	Meaning
R	the rank of tensor decomposition
l	the level index
R_l	the rank at l-th level
$\mathbf{v}^l$	the tensor-based user embedding at l-th level
$\mathbf{P}^l$	the projection matrix that maps from the l-th level patterns to the patterns at the $(l-1)$-th level
$\mathcal{X}$, $\mathbf{X}$, $\mathbf{x}$, x	Tensor, matrix, vector, scalar
$\mathbf{X}_{(i,j)}$	the scalar at the $\{i, j\}$ position of matrix $\mathbf{X}$
$\mathbf{X}_{(r,:)}$	the r-th row of matrix $\mathbf{X}$
$\mathbf{X}_{(m)}$	the mode-m unfolding of tensor $\mathcal{X}$
$\mathbb{U}$	the set of factor matrices
$\mathbf{U}_c^{(m)}$	the m-th mode factor matrix from the tensor of class c
$\mathbf{U}_c^l$	the l-th level factor matrix from the tensor of class c
I_1,..., I_M	the dimensionality of mode 1, ..., M

and matrix factorization in the context of experts recommendation in multiple areas simultaneously. However, there is limited research that discovers hierarchical nested subspace in the tensor subspace [31]. Özdemir et al. [31] construct a data-dependent multi-scale subspace to better represent the data. To do so, the authors first construct a tree structure by partitioning the tensor into a collection of permuted sub-tensors, and then construct the multi-scale subspace by applying HoSVD to each sub-tensor.

Summary. The existing predictive analytics on MOOCs considers various ways of constructing a matrix-based feature space. We argue that tensors could be a more suitable representation for student behavioral modeling due to their flexibility in representing the multi-way interaction of the behavioral data. In this regard, Sahebi et al. [39] have shown success in using a tensor-based approach to model the students' learning process, and predict student performance. However, the multi-way interactions as behavior patterns have not been discussed, and a more interpretable pattern discovery that can support a comprehensive understanding of student behaviors is missing. On the other hand, most hierarchical tensor factorization methods tend to recursively decompose the tensor modes by a pre-specified dimension tree [3, 29]. Our work, instead, is closer to the multi-level tensor factorization approach by [32], which recursively factorizes the residual tensors to obtain a multi-level representation of the subspace. However, to the best of our knowledge, there has been no work yet that discovers the common and discriminative patterns at multiple levels, especially in the area of predictive modeling in educational data mining.

3 PROBLEM FORMULATION

In this section, we start with a brief introduction to tensor notions and operations and then formulate the problem considered in this study. Table 1 summarizes the notations used in this paper.

3.1 Preliminaries

3.1.1 Tensors. A tensor is a multidimensional array. Let x denote a scalar, $\mathbf{x}$ a vector, $\mathbf{X}$ a matrix, and then $\mathcal{X}$ is the extension of these concepts to higher dimensions. The *order* of the tensor is the number of modes (or ways) in the tensor.

3.1.2 Tensor Operations. The basic tensor operations that we use in this study are:

Mode-n unfolding, or matricization, is the process of re-ordering elements of tensor $\mathcal{X}$ to form a matrix $\mathbf{X}_{(n)}$, where n is the dimension index upon which the unfolding performs. In the case of the three-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, its matricization in the first mode can be denoted as $\mathbf{X}_{(1)} \in \mathbb{R}^{I_1 \times (I_2 \times I_3)}$.

CP Decomposition [15] or CANDECOMP/PARAFAC decomposition expresses an m-way tensor $\mathcal{X}$ of size $I_1 \times \cdots \times I_M$ as $\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \cdots \circ \mathbf{u}_r^{(M)}$, where $\mathbf{u}_r^m \in \mathbb{R}^{I_m}$ for $m = 1, \ldots, M$ and $r = 1, \ldots, \mathbf{R}$. By using "kruskal operator," [24], we can shorthand the CP decomposition as $\mathcal{X} \approx \left[\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(M)}\right]$, where the factor matrix $\mathbf{U}^m$ is defined as $\mathbf{U}^m = \left[\mathbf{u}_1^{(m)} \cdots \mathbf{u}_R^{(m)}\right]$.

3.2 Problem Formulation with Tensors

To motivate the problem in the context of a real-world dataset, we discuss the application of NMF, Non-negative Tensor Factorization (NTF), discriminative NTF, and hierarchical NTF with a toy dataset. This dataset was extracted from one of the most popular courses in the XueTangX dataset (full dataset details in Section 5.1). Each student event, or *activity*, is associated with three attributes: *time* (d_1, d_2), *source* (s_1, s_2), and *type* ($t_1, t_2, \ldots, t_7$).

3.2.1 NMF. Let matrix $\mathbf{X}$ denote the aggregated activities that users have been recorded engaging in on the XueTangX platform for this course. The nature of matrix $\mathbf{X}$ is a two-dimensional array, which restricts its capability of integrating further information [38]. In this way, we can either drop one of the attributes or force the third dimension to be combined with the second dimension. Figure 2(a) shows the case where $\mathbf{X}'$ contains only *source*×*type*, and Figure 2(b) shows the case where $\mathbf{X}''$ contains *source*×(*type*+*type*), where (*type*+*type*) can be considered as a repeated vector to jointly represent the event activity and the day.

With the behavior described by $\mathbf{X}$, the bottom part of Figure 2(a-b) show the respective low-rank factor matrices approximated by NMF, the *source* factors (left), and the *type* factors (right), since they provide the low-dimensional representations of each source and each activity, respectively. Compared to $\mathbf{X}'$, $\mathbf{X}''$ has the additional advantage of revealing the low-dimensional representation of each activity on different days.

3.2.2 NTF. Alternatively, we can use a tensor to represent the same dataset (Figure 2(c)). With the given data of a ternary relation nature [43], we could use a third-order tensor $\mathcal{X}$ to denote a *source*×*day*×*type* activity.

NTF techniques can be applied to obtain three low-dimensional representations: *source* factors, *day* factors, and *type* factors, as $\mathcal{X} \approx \left[\mathbf{S}, \mathbf{D}, \mathbf{T}\right]$. As a result, each pattern comes with a set: a between-activity vector $\mathbf{t}$ to describe the activity dynamics; a between-source vector $\mathbf{s}$ to describe the usage tendency between different sources; and an across-day vector $\mathbf{d}$ to describe the temporal dynamics (e.g., p_1 in Figure 2(c)). Compared to factorizing the unfolding matrix $\mathbf{X}''$ (Figure 2(b)), NTF introduces the day-specific factors. This significantly increases the presentation capability of the patterns by revealing a more direct across-day (temporal) dynamics. Each pattern now represents the interplay of three factors, describing the tendency from different perspectives. With the rich attributes in the behavioral dataset on MOOCs, we thus use tensors to model

the behaviors, with the hope that doing so can provide behavioral patterns with the interpretations from different aspects.

3.2.3 Discriminant NTF. Standard NTF provides meaningful patterns for simultaneously analyzing the behaviors from multiple aspects. This substantially increases the capability of studying and interpreting the behaviors on MOOC platforms with rich dynamics. However, this still does not sufficiently serve the desire to comprehensively understand and investigate student behaviors on these platforms. For example, one of the most interesting questions is which behavior patterns are shared by *completed* students and *dropout* students, and which differentiate the two groups (Figure 2(d)). Through understanding the commonality and differences, researchers can better design course interactions and content to help more students successfully complete the course.

One could easily use NTF to fit the behavior tensors from each group of users, separately. However, this approach does not take advantage of any shared behavior patterns between the two groups. As the behavior moves to high-dimensional tensor space, this could potentially lead to the under-fitting problem. Besides, with patterns generated for each data tensor separately, it needs to perform an additional post-hoc analysis to determine common and discriminative patterns. This is a non-trivial attempt to align the common and discriminative patterns, in the case of each pattern being represented by multiple vectors from different aspects. In this regard, discriminative NTF is set to jointly factorize the tensors constructed from different groups of users with the following objective considering CP decomposition:

$$\mathcal{L}_{\text{disc}} = \left\|\mathcal{X}_{\text{Completed}} - \left[\mathbf{S}, \mathbf{D}, \mathbf{T}\right]\right\|^2 + \left\|\mathcal{X}_{\text{Dropout}} - \left[\mathbf{S}', \mathbf{D}', \mathbf{T}'\right]\right\|^2 + \Omega(\mathbf{S}, \mathbf{D}, \mathbf{T}, \mathbf{S}', \mathbf{D}', \mathbf{T}'), \tag{1}$$

where $\Omega(\cdot)$ is the function to promote the simultaneous discovery of the common and discriminative patterns [22, 47].

3.2.4 Hierarchical NTF. Previous works on NTF or discriminative NTF for unsupervised pattern discovery focus on finding a set of patterns at equal granularity, or in a flat structure. Although they are adequately expressive to reveal the behavior dynamics from different aspects, they can not provide the relations between patterns (such as parent-child and sibling relations).

A hierarchical non-negative tensor factorization (HierNTF) is more desirable than a set of "flat" patterns, because one can work with the pattern exploration in a hierarchy. As opposed to going through each pattern individually, this results in more efficient pattern understanding. HierNTF can be analogous to hierarchical topic modeling, such as hierarchical Latent Dirichlet Allocation (hLDA) [10], where patterns at higher levels in the hierarchy present "abstract" behavior topics, and ones at lower levels reveal more "specific" behavior topics.

3.2.5 Problem Statement. Our problem falls into the combination of the discriminative NTF and HierNTF. We would like to identify common and discriminative patterns nested at multiple levels for a deeper understanding of the relationship between students' multi-way behavior dynamics and their course performance. Before we give the formulation of the studied problem, we would like to first clarify some basic concepts used later.

Definition 3.1. *Individual Behavior Tensor.* Let $\mathcal{X}^{(u)} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_M}$ be an M-way tensor representing an individual user u, with each entry in the $\mathcal{X}^{(u)}$ being an activity from user u that is jointly described by M attributes.

The attributes can be data or platform dependent, such as a time-varying attribute tensor constructed from demographic and behavior attributes associated with users at different time stamps [36]. The individual behavior tensor can be considered as a multi-way representation of each student. With that, for each performance group, we can compute the *collective behavior tensor*.

Definition 3.2. *Collective Behavior Tensor.* Let $\mathcal{X}_c \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_M}$ be an M-way tensor that users M attributes to describe the collective activities from a group of users indexed by c.

Students at each performance group can be jointly represented by a collective behavior tensor that captures the full multi-way feature interactions of their activities. Then, we can combine the tensors for the two opposite performance groups to construct the *coupled tensors*.

Definition 3.3. *Coupled Tensors.* Coupled Tensors $\boldsymbol{\mathcal{X}} = \{\mathcal{X}_c\}$ is a pair of tensors with identical attributes, i.e., $I_1^c = I_1^{\bar{c}}, I_2^c = I_2^{\bar{c}}, \ldots, I_M^c = I_M^{\bar{c}}$, where c is the index of user group that tensor $\mathcal{X}$ is constructed from and $\bar{c}$ represents the counterpart class.

The coupled tensors $\boldsymbol{\mathcal{X}}$ can be constructed in various ways, depending on the performance metric selected, such as $c \in$ {dropouts, completion} or $c \in$ {certificates, no-certificates}. Inspired by [35], we construct the coupled tensor as follows:

$$\mathcal{X}_c = \frac{1}{|U_c|} \sum_{u \in U_c} \mathcal{X}^{(u)}, \tag{2}$$

where U_c is the subset of users u with u belonging to class c. While each individual behavior tensor captures the full-order feature interactions explained by her activities, the full interplay between the M attributes for each group of students can be contained within the tensor structure corresponding to the group.

Definition 3.4. *Multi-way Behavior Pattern.* A multi-way behavior pattern is a a collection of M vectors $(x^{(1)}, x^{(2)}, \ldots, x^{(M)}), M \geq 2$, where $x^{(m)} \in \mathbb{R}^{I_m}$ is a vector to describe the pattern with the m-th attribute.

Definition 3.5. *Pattern Hierarchy.* Let $\mathbf{P}^l \in \mathbb{R}^{R_l \times R_{l-1}}$ denote a pattern hierarchy that specifies the relationships between the behavior patterns in the consecutive levels, i.e., level l and level $l-1$, where $1 \leq l \leq L$. $\mathbf{P}^l$ can be considered as a projection matrix that maps the pattern from the l-th level to the ones at the $(l-1)$-th level.

Definition 3.6. *Tensor-based User Embedding.* A tensor-based user embedding $v \in \mathbb{R}^R$ is the student's vector representation that preserves each student's behavior tensor with a lower-dimensional feature space $\mathbb{R}^R$, given the tensor's rank R.

With the definitions above, we define the iterative common and discriminative pattern analysis as follows:

Problem 1. Given a set of individual behavior tensors $\mathcal{X}^{(u)} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_M}, (u = 1, 2, \ldots, N)$ corresponding to C categories, $C = 2$, and a set of unseen test data $\mathcal{X}^{(t)} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_M}, (t = 1, 2, \ldots, T)$, our goal is to iteratively identify a set of patterns that reveal the

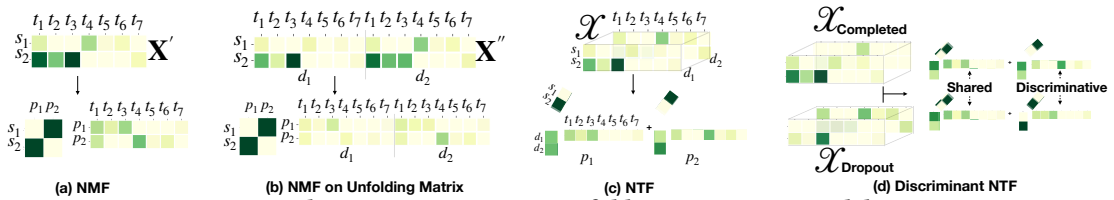

Figure 2: Comparison between NMF, NMF on unfolding Matrix, NTF and discriminant NTF.

common and discriminative behaviors at multiple levels, and then use the learned patterns as bases to infer the user embeddings for prediction of the group membership from the unseen students.

Specifically, the task is threefold: (1) collective behavior pattern inference for discovering the common and discriminative patterns; (2) iterative pattern discovery at multiple levels with hierarchy; and (3) embedding projection for test samples based on the iterative patterns for classification. In other words, the first two tasks are aimed at revealing interpretable patterns that could explain the interplays between the behavior attributes, and the last task is to discover the relationship between student performance and the multi-way patterns.

4 SOLUTIONS

In this section, we introduce an iterative tensor factorization method named `iDisc` for the coupled tensors, $\boldsymbol{\mathcal{X}} = \{\mathcal{X}_c\}$. Figure 3 illustrates the overview of `iDisc`. There are two-stages: (1) iterative application of discriminative tensor subspace learning; and (2) representation learning for the unseen student based on the multi-level patterns.

4.1 Iterative Discriminative Tensor Subspace Learning

This component iteratively applies the following two-step approach: (1) discriminant low-rank tensor approximation, followed by (2) computing and passing the residual tensor into the next level.

4.1.1 Discriminant Tensor Factorization. Conventional tensor factorization seeks a set of N factor matrices $\left[\mathbb{U}\right] = \left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}\right]_c^l$ from a behavior tensor $\mathcal{X}_c^l$ at level l for class c. One such example is:

$$\mathcal{L}^l = \underbrace{\sum_c \left\| {\mathcal{X}_c}^l - \left[\mathbb{U}\right]_c^l \right\|^2}_{\text{Loss for Coupled Tensors Factorization}} . \tag{3}$$

Through Eq. 3, we could obtain a set of independent factor matrices (or behavior patterns) for each of the performance groups, respectively. However, this does not consider the commonality and differences among the coupled tensors.

In order to take the commonality between the two behavior tensors into consideration and allow discrimination against each other, we introduce two sets of auxiliary tensors $\mathcal{S}_c^l$ and $\mathcal{Z}_c^l$ that capture the shared behavior and the discriminative behaviors among the coupled tensors. Inspired by [47], $\mathcal{S}_c^l$ and $\mathcal{Z}_c^l$ are computed based on the coupled tensors $\boldsymbol{\mathcal{X}}$ with the clamping function (Eq. 7 and Eq. 9 in [47]). The rational behind the auxiliary tensors is that; we would like to have discriminative tensors that contain only the unique signals for each class, and common tensors to hold what is shared among the coupled tensors. A collective tensor factorization framework is then leveraged to jointly factorize the coupled tensors and the auxiliary tensors as follows:

$$\begin{aligned}\mathcal{J}^l = \mathcal{L}^l + \underbrace{\lambda_0\Big(\sum_c \left\| {\mathcal{Z}_c}^l - \left[\mathcal{W}_{\mathcal{Z}}; \mathbb{U}\right]_c^l \right\|^2 + \sum_c \left\| {\mathcal{S}_c}^l - \left[\mathcal{W}_{\mathcal{S}}; \mathbb{U}\right]_c^l \right\|^2\Big)}_{\text{Loss for Auxiliary Tensor Factorization}} \\ + \underbrace{f\left(\mathbf{U}_c^l, \mathbf{U}_{\bar{c}}^l, \mathcal{W}_{\mathcal{S}}^l\right)}_{\text{Loss for Pattern Alignment}} + \underbrace{g\left(\mathbf{U}_c^{l-1}, \mathbf{U}_c^l, \mathbf{P}^l\right)}_{\text{Loss for Pattern Hierarchy}} + \underbrace{h\left(\mathbf{P}^l\right)}_{\text{L1 penalty}} \\ s.t. \left\|\mathbf{U}_c^l\right\|_2 = 1, \forall\, c,\end{aligned} \tag{4}$$

where:

- $\mathcal{W}_{\mathcal{Z}}$ and $\mathcal{W}_{\mathcal{S}}$ are the core tensors with super diagonal entries;
- $f\left(\cdot\right)$ is the function to enforce the similar components to be aligned correspondingly and defined as:

$$f\left(\mathbf{U}_c^l, \mathbf{U}_{\bar{c}}^l, \mathcal{W}_{\mathcal{S}}^l\right) = \lambda_1 \sum_m \left(\left\| \text{diag}\left(\mathcal{W}_{\mathcal{S}q}\right)\mathbf{U}_c^m - \text{diag}\left(\mathcal{W}_{\mathcal{S}\bar{c}}\right)\mathbf{U}_{\bar{c}}^m \right\|^2 \right); \tag{5}$$

- $g\left(\cdot\right)$ is the function that learns the shared mapping $\mathbf{P}^l$ by the coupled tensors, between the patterns at the consecutive levels. We can consider this operation as performing a matrix decomposition from $\mathbf{U}_c^{l-1}$ to $\mathbf{U}_c^l$ and $\mathbf{P}^l$ as:

$$g\left(\mathbf{U}_c^{l-1}, \mathbf{U}_c^l, \mathbf{P}^l\right) = \lambda_2 \left\| \mathbf{U}_c^{l-1} - \mathbf{U}_c^l \mathbf{P}^l \right\|^2, \tag{6}$$

  assuming that we already have the values of $\mathbf{U}_c^{l-1}$ for l-th level pattern discovery;
- $h\left(\cdot\right)$ is an $L1$ penalty function that encourages sparsity in $\mathbf{P}^l$ to promote the exclusive mapping between the factor matrices at the consecutive levels. Considering a more interpretable pattern hierarchy, we use $L1$-norm, since it can function as a proxy for the $L0$ norm, to minimize the number of nonzero elements while maintaining the convexity of the cost function when estimating P the others fixed. In this manner, we ensure the higher-level patterns are mapped to exclusive lower-level ones; and
- $\lambda_0, \lambda_1, \lambda_2, \lambda_3$ are the respective parameters to weigh in each objective.

With Eq. 4, common and discriminative patterns discovery at the l-th level becomes an optimization objective as:

$$\theta^l = \text{argmin}_\theta^l \; \mathcal{J}^l, \tag{7}$$

where $\theta^l = \{\mathbb{U}, \mathcal{W}_{\mathcal{Z}}, \mathcal{W}_{\mathcal{S}}, \mathbf{P}\}_c^l$.

4.1.2 Obtain the residual tensors. Once θ^l is determined, the reconstructed tensor can be obtained by:

$$\hat{\mathcal{X}}_c^l \approx \left[\mathbb{U}_c^l\right], \tag{8}$$

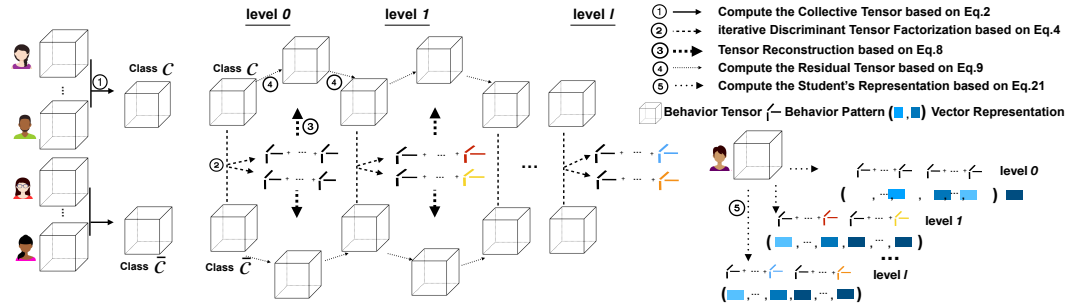

Figure 3: The overview of iDisc's workflow.

and therefore the residual tensor can be computed as:

$$\mathcal{E}_c^l = \mathcal{X}_c^l - \hat{\mathcal{X}}_c^l, \forall\, c. \tag{9}$$

Let $\mathcal{X}_c^{l+1}$ denote the tensors for the identification of common and discriminative patterns at the next level $l+1$. We first obtain $\mathcal{X}_c^{l+1}$ as: $\mathcal{X}_c^{l+1} = \mathcal{E}_c^l$, where $\mathcal{E}_c^l$ is the residual tensor as aforementioned. With $\mathcal{X}_c^{l+1}$, we can further identify the common and discriminative patterns at level $l+1$ with Eq. 4.

Algorithm 1: Algorithm iDisc for discovering the shared and discriminative subspace from coupled tensors at multiple levels.

Input : Coupled tensors $\mathcal{X}$, $\{R\}_l^L$, and $\{\lambda_0, \lambda_1, \lambda_2, \lambda_3\}$
Output: $\{\theta\}_l^L$, where $\theta = \{\mathbb{U}, \mathcal{W}_\mathcal{Z}, \mathcal{W}_\mathcal{S}, \mathbf{P}\}_c$

```
Let l = 0;
while l < L do
    Construct {Z_c}^l and {S_c}^l from {X_c}^l based on Eq.7 and Eq.9 in [47];
    Obtain θ^l by solving Eq. 4;
    Reconstruct {X_c}^l based on Eq. 8;
    Compute E_c^l based on Eq. 9;
    Let l = l + 1;
    Let X_c^l = E_c^l
end
```

4.1.3 Parameter Optimization of iDisc. For simplicity of notation, we omit the level l in all notations since the optimization is performed per level with the focus on the unknown θ^l and the θ^{l-1} are learned at level $l-1$. We also omit the mode notation m because all modes share the same optimization process. Let $\mathbf{U}_c$ represent the mode-m factor matrix at level-l, instead of the rather complex form of $\{\mathbf{U}_c^{(m)}\}_l$. We use $\overline{\mathbb{U}}_c$ to denote the set of factor matrices for $\mathcal{X}_c$ that correspond to modes other than m, and $\bar{c}$ to denote the class that is not c.

Since objective function $\mathcal{J}$ is not convex with respect to θ, we aim to find a local minimum for $\mathcal{J}$ by iteratively updating each in θ.

1. Update $\mathbf{U}_c$*, fix others.* The optimization of $\mathbf{U}_c$ is equivalent to the following least squares loss functions [47]:

$$\begin{aligned}\mathbf{U}_c \leftarrow \underset{\mathbf{U}_c \geq 0}{\operatorname{argmin}} \frac{1}{2}\Big(\frac{1}{n_c}\Big\|\mathbf{X}_c - \mathbf{U}_c\,(\odot\overline{\mathbb{U}}_c)^T\Big\|_F^2\Big) \\ + \lambda_0\Big(\Big\|\mathbf{Z}_c - \mathbf{U}_c\Lambda_{\mathbf{w}_{Z_c}}(\odot\overline{\mathbb{U}}_c)^T\Big\|_F^2 + \Big\|\mathbf{S}_c - \mathbf{U}_c\Lambda_{\mathbf{w}_{S_c}}(\odot\overline{\mathbb{U}}_c)^T\Big\|_F^2\Big) \\ + \lambda_1\Big\|\mathbf{U}_c\Lambda_{\mathbf{w}_{S_c}} - \mathbf{U}_{\bar{c}}\Lambda_{\mathbf{w}_{S_{\bar{c}}}}\Big\|_F^2 + \lambda_2\Big\|\mathbf{U}_c^{l-1} - \mathbf{U}_c\mathbf{P}\Big\|^2,\end{aligned} \tag{10}$$

where $\mathbf{X}_c$ is the mode-m unfolding of tensor $\mathcal{X}_c$. Then the gradient update of $\mathbf{U}_c$ can be computed as:

$$\begin{aligned}\nabla_{\mathbf{U}_c}\mathcal{J} = \frac{1}{n_c}\Big(\mathbf{U}_c(\odot\overline{\mathbb{U}}_c)^T - \mathbf{X}_c\Big)(\odot\overline{\mathbb{U}}_c) \\ + \lambda_0\Big(\Big(\mathbf{U}_c\Lambda_{\mathbf{w}_{Z_c}}(\odot\overline{\mathbb{U}}_c)^T - \mathbf{Z}_c\Big)(\odot\overline{\mathbb{U}}_c)\Lambda_{\mathbf{w}_{Z_c}}^T + \Big(\mathbf{U}_c\Lambda_{\mathbf{w}_{S_c}}(\odot\overline{\mathbb{U}}_c)^T - \mathbf{S}_c\Big)(\odot\overline{\mathbb{U}}_c)\Lambda_{\mathbf{w}_{S_c}}^T\Big) \\ + \lambda_1\Big(\mathbf{U}_c\Lambda_{\mathbf{w}_{S_c}} - \mathbf{U}_{\bar{c}}\Lambda_{\mathbf{w}_{S_{\bar{c}}}}\Big)\Lambda_{\mathbf{w}_c} - \lambda_2\mathbf{P}\Big(\mathbf{U}_c^{l-1} - \mathbf{U}_c\mathbf{P}\Big).\end{aligned} \tag{11}$$

2. Update $\mathcal{W}_{Z_c}$*, fix others.*
Let $\mathbf{w}_{Z_c}$ denote $\mathbf{w}$ for tensor $\mathcal{X}_c$. The optimization of $\mathbf{w}_{Z_c}$ is equivalent to the following problem:

$$\mathbf{w}_{Z_c} \leftarrow \underset{\mathbf{w}_c \geq 0}{\operatorname{argmin}}\ \lambda_0\Big\|\mathbf{Z}_c - \Lambda_{\mathbf{w}_{Z_c}}(\odot\mathbb{U}_c)^T\Big\|^2, \tag{12}$$

where $\Lambda_{\mathbf{w}_{Z_c}} \in \mathbb{R}^{R\times R\times R}$ is the tensor with $\mathbf{w}_{Z_c}$ as its super-diagonal entries. The gradient update of $\mathbf{w}_c$ is:

$$\nabla_{\mathbf{w}_{Z_c}}\mathcal{J} = \lambda_0\Big(\Lambda_{\mathbf{w}_{Z_c}}(\odot\mathbb{U}_c)^T - \mathcal{Z}_c\Big)(\odot\mathbb{U}_c). \tag{13}$$

3. Update $\mathcal{W}_{S_c}$*, fix others.*
Let $\mathbf{w}_{S_c}$ denote $\mathbf{w}$ for tensor $\mathcal{X}_c$. The optimization of $\mathbf{w}_{S_c}$ is equivalent to the following problem:

$$\mathbf{w}_{S_c} \leftarrow \underset{\mathbf{w}_{S_c} \geq 0}{\operatorname{argmin}}\ \lambda_0\Big\|\mathbf{S}_c - \Lambda_{\mathbf{w}_{S_c}}(\odot\mathbb{U}_c)^T\Big\|^2 + \lambda_1\Big\|\mathbf{U}_c\Lambda_{\mathbf{w}_{S_c}} - \mathbf{U}_{\bar{c}}\Lambda_{\mathbf{w}_{S_{\bar{c}}}}\Big\|^2. \tag{14}$$

The gradient update of $\mathbf{w}_{S_c}$ can be derived as:

$$\nabla_{\mathbf{w}_{S_c}}\mathcal{J} = \lambda_0\Big(\Lambda_{\mathbf{w}_{S_c}}(\odot\mathbb{U}_c)^T - \mathcal{S}_c\Big)(\odot\mathbb{U}_c) - \lambda_1\Big(\mathbf{U}_c{}^T\Big(\Lambda_{\mathbf{w}_{S_c}}\mathbf{U}_c - \Lambda_{\mathbf{w}_{S_{\bar{c}}}}\mathbf{U}_{\bar{c}}\Big)\Big) \tag{15}$$

4. Update $\mathbf{P}$*, fix others.* The optimization of $\mathbf{P}$ is equivalent to a co-regularized collective matrix factorization problem with sparsity constraints [6, 41]:

$$\mathbf{P} \leftarrow \underset{\mathbf{P}\geq 0}{\operatorname{argmin}}\ \lambda_2\Big(\frac{1}{2}\Big\|\mathbf{U}_c^{l-1} - \mathbf{U}_c\mathbf{P}\Big\|^2 + \frac{1}{2}\Big\|\mathbf{U}_{\bar{c}}^{l-1} - \mathbf{U}_{\bar{c}}\mathbf{P}\Big\|^2\Big) + \lambda_3\|\mathbf{P}\|_1. \tag{16}$$

Since the sparsity is applied to each row of $\mathbf{P}$, each row $\mathbf{P}_{(r,:)}$ can be updated based on the following gradient:

$$\nabla_{\mathbf{P}_{(r,:)}}\mathcal{J} = -\lambda_2\Big(\mathbf{U}_{c\,(r,:)}^{T}(\mathbf{U}_{c}^{l-1} - \mathbf{U}_{c\,(:,r)}\mathbf{P}_{(r,:)}) + \mathbf{U}_{\bar{c}\,(r,:)}^{T}(\mathbf{U}_{\bar{c}}^{l-1} - \mathbf{U}_{\bar{c}\,(:,r)}\mathbf{P}_{(r,:)})\Big) + \lambda_3 \mathrm{sgn}(\mathbf{P}_{(r,:)}). \tag{17}$$

The details of iDisc are summarized in Alg. 1 [1].

4.1.4 Time Complexity Analysis. The time complexity is mainly consumed by updating each factor matrix $\mathbf{U}_c$ in iDisc from computing $\nabla_{\mathbf{U}_c}\mathcal{J}$. From Eq. 11, we need to compute $\mathbf{U}_c(\odot\overline{\mathbb{U}}_c)^T(\odot\overline{\mathbb{U}}_c)$ and $\mathbf{X}_c(\odot\overline{\mathbb{U}}_c)$ in the first term. Note $\mathbf{U}_c \in \mathbb{R}^{I_m \times R}$ and $\mathbf{X}_c \in \mathbb{R}^{I_m \times \prod_{i\neq m} I_i}$ and therefore we have $\mathbf{U}_c(\odot\overline{\mathbb{U}}_c)^T(\odot\overline{\mathbb{U}}_c) \in \mathbb{R}^{I_m \times R}$ and $\mathbf{X}_c(\odot\overline{\mathbb{U}}_c) \in \mathbb{R}^{I_m \times R}$. The operation of matricized tensor times Khatri-Rao product ($\mathbf{X}_c(\odot\overline{\mathbb{U}}_c)$) is often considered a bottleneck for CP decomposition due to the expensive computational cost [48]. In practice, the sparsity of the tensor is leveraged for an efficient computation for this operation [2, 48]. Particularly, the complexity can be reduced by only considering the computation for nonzero observations in $\mathcal{X}_c$. Let x_h denote the h-th nonzero observation in $\mathcal{X}_c$ and its subscripts in $\mathcal{X}_c$ as $(I_{1_h}, I_{2_h}, \ldots, I_{M_h})$. If there are H non-zeros, i.g, $H = nnz(\mathcal{X}_c)$, we would just need an H-vector to store the real values of $\mathcal{X}_c$. In this case, the element-wise computation for $\mathbf{X}_c(\odot\overline{\mathbb{U}}_c)$ can be written as:

$$\Big(\mathbf{X}_c(\odot\overline{\mathbb{U}}_c)\Big)_{(i,r)} = \sum_{\substack{h=1 \\ I_{m_h}=i}}^{H} x_h \prod_{\substack{m'=1 \\ m'\neq m}}^{M} \mathbf{U}^{(m')}_{(I_{m'_h},r)}, \quad \text{for } i = 1, \ldots, I_m \text{ , and } r = 1, \ldots, R, \tag{18}$$

where the computation of Khatri-Rao product can be ignored when $x_h = 0$. Therefore, the time complexity for computing $\mathbf{X}_c(\odot\overline{\mathbb{U}}_c)$ for each mode per iteration is $O(nnz(\mathcal{X}_c)I_m R)$. The element-wise of $\mathbf{U}_c(\odot\overline{\mathbb{U}}_c)^T(\odot\overline{\mathbb{U}}_c)$ can be efficiently computed as [1, 42]:

$$\Big(\mathbf{U}_c(\odot\overline{\mathbb{U}}_c)^T(\odot\overline{\mathbb{U}}_c)\Big)_{(i,r)} = \sum_{j=1}^{R}\left(\mathbf{U}^{(m)}_{(i,j)} \prod_{\substack{m'=1 \\ m'\neq m}}^{M} \sum_{i=1}^{I_{m'}} \mathbf{U}^{(m')^T}_{(i,j)} \mathbf{U}^{(m')}_{(i,r)}\right). \tag{19}$$

Therefore, the time complexity as $O(\hat{I}\,R^2)$, where $\hat{I} = \sum_{m'=1}^{M} I_{m'} - I_m$ and the overall time complexity for the above two terms is $O(HI_mR + \hat{I}\,R^2)$. Similarly, the time complexity for terms involving tensors $\mathcal{Z}$ and $\mathcal{S}$ becomes $O(H' I_m R^2 + \hat{I}\,R^3)$ due to the additional loop introduced by the weight vector $\mathbf{w}$, where H' is the respective nonzero observations in the auxiliary tensors. Since $H' \leq H$, with $M \ll H$, $R \ll H$, and $\hat{I} \ll H$, we can see that the running time is expected to scale linearly with the number of nonzero observations in $\mathcal{X}_c$.

4.2 Embedding Learning for the Unseen Student

This section explains the inference of the student's embedding in the latent space anchored by the factor matrices. The individual behavior tensor $\mathcal{X}_t$ for unseen student t with an unknown class is constructed based on his or her logs with the system. The mode settings of tensor $\mathcal{X}_t$ are the same as for the collective behavior tensors (e.g., $\mathcal{X}_c$). With the students' individual tensor and the factor matrices, we follow iDisc to first obtain the corresponding auxiliary $\mathcal{Z}$ tensors at l-th level, $\mathcal{Z}^l_{t_{\hat{c}}}, \forall\, \hat{c} \in \{c, \bar{c}\}$, for student t, by following the clapping function in Eq. 7 in [47]. Then, the equation to compute the embedding becomes:

$$\mathbf{v}^l_{t_{\hat{c}}} = \mathcal{Z}^l_{t_{\hat{c}}} \times_m \left\{\mathbf{U}^{(m)}\right\}^l_{\hat{c}}, \forall\, \hat{c} \in \{c, \bar{c}\}, \tag{20}$$

which follows a typical computation of the core tensor, given the data tensor and its factor matrices.

It is worth noting that since PARAFAC decomposition does not enforce the orthogonal property in each of the factor matrices, direct computation of Eq. 20 is not feasible. Recent work by [8] proposes an efficient estimation of the core tensor for PARAFAC decomposition. By following the algorithm 1 in [8] to efficiently estimate $\mathbf{v}^l_{c_t}$ and then the embedding of student s at l-th level $\mathbf{v}^l_t \in \mathbb{R}^{2R_l}$ can be obtained by:

$$\mathbf{v}^l_t = \left[\mathbf{v}^l_{t_c} | \mathbf{v}^l_{t_{\bar{c}}}\right]. \tag{21}$$

5 EXPERIMENTS

In this section, we conduct systematic experiments to evaluate the quality of iDisc. In following, we first describe the data used for the experiments. The content of the rest of this section is structured to answer the following questions:

- Can iDisc reveal meaningful patterns?
- How does iDisc perform in comparison with state-of-art methods from predictive modeling in learning analytics?
- Can iDisc scale for the dataset at a massive scale?

5.1 Data

We experiment with nine courses/sessions from three publicly-available MOOC platforms: edX, ASSISTments, and XueTangX. The statistics of the datasets are provided in Table 2.

Table 2: Dataset and Tensor Modes Description.

Dataset	Course	#Users	#Activity	#Areas	#Education	Stay	
edX	Course A	57,715	-				
	Course B	66,731	-	34	5	5	
	Course C	169,621	-				
				#Days	#Events	#Source	
XueTangX	Course 1	12,004	652,701				
	Course 2	10,321	877,805	14	7	2	
	Course 3	9,382	907,118				
				#Problems	#KC	#Views	#Duration
ASSISTments	Year 2004	912	580,785	376	58	7	10
	Year 2005	2,392	521,751	266	59	4	10
	Year 2006	2,584	686,868	409	69	4	10

edX. The edX [33] dataset is comprised of de-identified data from "Introduction to Computer Science" (Fall 2012, Spring 2013, and Summer 2013) from MITx (Course A and B in Table 2) and HarvardX (Course C). This dataset does not have detailed event logs. However, the data are aggregated records, where each record represents the summary statistics for one individual's activity in the edX course with her demographic information. We select the three most popular courses from this dataset. For this dataset, we construct a 34x5x5 tensor as *Area*×*Education*×*Stay*. Our goal is to predict whether the course completion certificate is earned by a student at the end of the course.

[1] Code is available at https://github.com/picsolab/iDisc

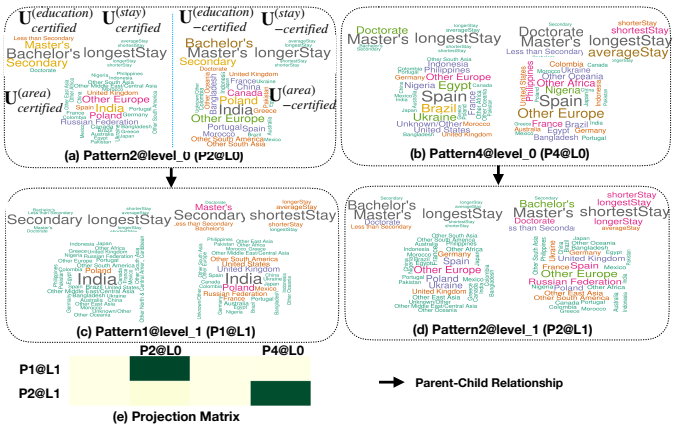

Figure 4: Output generated by `iDisc` on the MITx dataset. (e) shows the relationships between patterns in the first two levels; (a-d) show the associated patterns in the hierarchy.

XueTangX. XueTangX [50] is one of the largest MOOC platforms in China. The full dataset includes 79,186 students enrolled in 39 classes. Each enrollment is associated with a log of the student's activities, including watching lecture videos, working on course problems, accessing course modules, and so on. In total, there are 8,157,277 activity logs, and the longest lifetime of enrollment is five weeks. We take the three most popular courses from this dataset. The dataset statistics are shown in Table 2. We use the first two weeks to learn the factor matrices by constructing a 14x7x2 tensor as *Day×EventType×EventSource*. The goal is to correctly predict course completion.

ASSISTments. ASSISTments [16] is an online tutoring system used by more than 50,000 students around the world [5]. On ASSITments, students attempt to solve problems and receive feedback on those attempts. To assist the learning process, each problem is also associated with multiple knowledge components. We take the public dataset of the Math course on ASSISTments over three years (2004, 2005, and 2006) and the dataset characteristics are shown in Table 2. For each dataset, we construct a four-way tensor as *Problem×KnowledgeComponent×ProblemView ×ActionDuration*. Our aim is to classify the students as over-performing students and under-performing students, in terms of error-rate [2].

5.2 Qualitative Examination of the Patterns

In this section, we use "Introduction to Computer Science" on MITx during Spring 2013 to illustrate the outputs of `iDisc`. We first qualitatively examine the patterns, as well as the pattern hierarchy generated and then explain the relationship between the patterns and the performance outcome.

5.2.1 Common and Discriminative Pattern Discovery. Figure 4 describes the outputs by `iDisc`, with $\{R\}_l^L = \{4, 2\}$ (rank-4 at the first level and rank-2 at the second level). Particularly, Figure 4(e) shows the project matrix $\mathbf{P}^1 \in \mathbb{R}^{2\times4}$ that represents the hierarchical relationships between the set of patterns at the first two levels, where darker colors indicate stronger associations. We observe that pattern #1 at level 1(P1@L1) is strongly associated with pattern #2 at level 0(P2@L0). This suggests that P1@L1 could be a child pattern for P2@L0. Similarly, we observe the parent-child relationship between P4@L0 and P2@L1. Since the projection matrix is shared by the factor matrices from the coupled tensors, it is important to note that each of figures 4(a) through 4(d) refers to both patterns for the certified and non-certified group (i.e. as shown left and right in Figure 4(a)).

[2] https://pslcdatashop.web.cmu.edu/help?page=terms#error_rate

Table 3: Regression results for course end performance. Explanatory variables are the students' embeddings that correspond to the patterns presented in Figure 4 (a-d) (e.g., $\mathbf{v}^0_{certified}(2)$ refers to the values in students' embedding vector v for P2@L0 from the certified group). *Note:* **:p<.05; ***:p<.01.

	Certified (M1)		Certified (M2)
$\mathbf{v}^0_{\text{certified}}(2)$	−0.058 (0.046)	$\mathbf{v}^1_{\text{certified}}(1)$	−0.282 (0.188)
$\mathbf{v}^0_{\text{certified}}(4)$	−0.004 (0.046)	$\mathbf{v}^1_{\text{certified}}(2)$	2.285^{***} (0.649)
$\mathbf{v}^0_{\text{-certified}}(2)$	-0.113^{**} (0.048)	$\mathbf{v}^1_{\text{-certified}}(1)$	-0.620^{***} (0.080)
$\mathbf{v}^0_{\text{-certified}}(4)$	0.026 (0.046)	$\mathbf{v}^1_{\text{-certified}}(2)$	0.733^{***} (0.061)

Due to the space limit, we only discuss the details of two sets of multi-way patterns as word clouds in Figure 4(a) for P2@L0 and Figure 4(c) for P1@L1. Figure 4(a) describes a subgroup of students that are from the most populated countries (e.g., the United States, India, and Poland, based on $\mathbf{U}^{(area)}$) with education background mostly from Secondary to Master's ($\mathbf{U}^{(education)}$), and most of whom tend to stay on the edX platform ($\mathbf{U}^{(stay)}$) for a relatively long period. While the pattern from the certified group of students shares almost identical distributions in the area and educational background, what makes them slightly different was primarily the time spent on the platform; certified students($\mathbf{U}^{(stay)}_{\text{certified}}$) spend relatively longer than un-certified students ($\mathbf{U}^{(stay)}_{\text{-certified}}$). The set of patterns in Figure 4(c) is the child patterns of the Figure 4(a). Compared to their counterpart in level 0, they primarily describe the students from Indian (although the area distribution from the un-certified group of students spans more countries ($\mathbf{U}^{(area)}_{\text{-certified}}$)) with more focus on the middle level of education background (e.g., Secondary($\mathbf{U}^{(education)}_{\text{certified}}$)). The difference in their length of stay on the platform was more prominent in this set of patterns, where certified students have the longest stays with the edX platform ($\mathbf{U}^{(stay)}_{\text{certified}}$), and un-certified students generally have the least ($\mathbf{U}^{(stay)}_{\text{-certified}}$).

5.2.2 Simpson's Paradox Revisited. We performed multivariate logistic regression analysis to identify the patterns that can explain students' variation in obtaining the certificate for each level (M1 for level 0 and M2 for level 1). The dependent variable is whether or not the users are certified at the end of the course. The explanatory variables include the students' embeddings for the aforementioned patterns in Figure 4(a-d) in each level (e.g., $\mathbf{v}^1_{\text{certified}}(1)$ refers to the embeddings corresponding to P1@L1$_{\text{certified}}$). The embeddings are standardized to facilitate comparison among different variables.

Table 3 shows the estimated coefficients for M1 and M2. The only significant variable in M1 is the embeddings $\mathbf{v}^0_{\text{-certified}}(2)$ ($\beta = -0.113$, $p < .05$). This suggests that users who have shown more

activities in line with pattern P2@L0$_{\text{-certified}}$ appear to have less chance to earn the certificate. M2 shows that both embeddings $\mathbf{v}^1_{\text{certified}}(2)$ ($\beta = 2.285, p < .01$) and embeddings $\mathbf{v}^1_{\text{-certified}}(2)$ ($\beta = 0.733, p < .01$) reveal a significant and positive effect towards earning the certificate, with $\mathbf{v}^1_{\text{certified}}(2)$ having a much larger effect size. On the other hand, M2 also shows a significant and negative effect of having a larger value in $\mathbf{v}^1_{\text{-certified}}(1)$ ($\beta = -0.620, p < .01$).

We can consider multi-way patterns as principal components in PCA that bridge the original feature interactions in the high-dimensional space and the associated *loadings*. In this case, we would expect students from one class to have higher loading scores associated with patterns that are extracted from the same class. The regression result in M1 shows that the un-certified group of students does have larger loading scores. However, that is true only in $\mathbf{v}^0_{\text{-certified}}(2)$ for P2@L0 . This is not surprising, because in level 0 the two sets of patterns are in fact very similar to each other, as shown in Figure 4. The results in M2 confirm this expectation, with significantly higher loading scores $\mathbf{v}^1_{\text{certified}}(2)$ for certified students and significant higher loading scores $\mathbf{v}^1_{\text{-certified}}(1)$ for un-certified students. Contrary to our expectation, the certified group of students also have higher loading scores in $\mathbf{v}^1_{\text{-certified}}(2)$ (although much lower than $\mathbf{v}^1_{\text{certified}}(2)$). This could be explained by our observation in Figure 1, where P2@L1 captures a cluster of highly-educated students from certain European areas. They have a much higher chance of obtaining the certificate, regardless of having the longest stay or shortest stay with the platform.

Summary. We qualitatively examine the outputs generated by `iDisc`. The results show interesting properties of the proposed method. Our model reveals common and discriminative patterns at each level with their relationship explained via the projection matrix. Our regression analysis first explains the discriminative capability of the students' embeddings based on this set of patterns. More importantly, the analysis validates that the students' embeddings can be used to measure the relationship between the performance outcome with multi-way patterns from `iDisc`.

5.3 Quantitative Comparison

In this section, we report the results from the quantitative experiments in comparison with existing work commonly used in predictive analytics. Specifically, we conduct a classification task, in which the goal is to predict the students' performance at the end of the course defined in Section 5.1.

5.3.1 Baselines. We include baselines that are commonly seen in the area of predictive analytics in educational data mining as:

- Raw. We use the raw activity counts each day as features to train classifiers for prediction. This is the most common approach in predictive modeling for MOOCs.
- LDA. Coleman et al. [4] use LDA to capture the temporal element of the behavior data. We first discover the latent behavior patterns from Raw features with a varying number of topics, and use the topic membership of each student for the classification task.
- LadFG [36]. As one of the most cited works in MOOC predictive modeling, LadFG is a latent dynamic factor graph model that finds a mapping from students' time-varying attribute tensor to the observed learning outcome. We only evaluate the performance of LadFG in XueTangX dataset because it is the only one of the three that contains the necessary temporal dynamics.
- Factorization machines (FM) [37]. Factorization machines have been proposed and successfully applied to recommendation and prediction tasks. As the factorization model projects the input feature space into a latent space, it enables the learning of more complex interactions between features. We first convert each dimension as dummy variables for each student, and then concatenate all dimensions as a wide feature matrix.

It is worth noting the recent use of Deep Neural Nets (DNN) and their variants have shown promising performance compared to conventional machine learning approaches (e.g., [21, 45, 49]). However, the lack of interpretability of these models prevents their further application in problems driven by both interpretations and performance gains. We also compare `iDisc` with the existing work on discovering the common and discriminative patterns from multi-way data.

- `SDCDNTF`. `SDCDNTF` extends [22] and learns the common and discriminative patterns with different ranks. The input to the model consists of the rank and the number of shared patterns, along with the coupled tensors.
- `PairFac`. `PairFac` [47] learns the common and discriminative patterns with different ranks. Comparing to `SDCDNTF`, it does not require the input of the split.

We would like to point out that standard tensor factorization could serve as another baseline to compare with, in which students or their class reside as one of the dimensions and the corresponding factor matrix can naturally become features for downstream prediction tasks. However, we did not include it for two reasons: first, because the aforementioned baselines work as inductive models, where unseen students can be predicted based on the learned parameters, while simple tensor factorization serves as a transductive model and only predicts for the students that are available in the factorization; and second, because student populations on MOOC platforms can be of any size, from small to very large, the efficiency of standard tensor factorization with a large dimension size could be a practical problem for its real-world application.

5.3.2 Experiment Settings. For each dataset, we draw a training set of students from each class with replacement, and then obtain the embeddings of the out-of-bootstrap students. For this set of students, we perform a five-fold cross-validation with a k Nearest Neighbors classifier. We conduct five independent trials of this experiment and report the average classification accuracy. We select accuracy since both the training and testing dataset are constructed in a way that each class has an equal amount of students. For `SDCDNTF`, we experiment with α, β and $\gamma \in \{10^{-5}, 10^{-4}, 10^{-3}\}$. Finally, we set α and β in the same range, and $R = 6$ for both `PairFac` and `SDCDNTF` and derive two versions of `SDCDNTF` using $K \in \{2, 4\}$. To make a fair comparison with `PairFac` and `SDCDNTF` , we use two-level pattern discovery with rank-4 and rank-2 in each level for `iDisc`, respectively. For LDA and FM, we experiment with a varying number of topics /factors and report the best performance. For LadFG, we keep the suggested parameters from their paper.

Table 4: Classification Results in Accuracy.

Dataset		edX Course A	edX Course B	edX Course C	XueTangX Course 1	XueTangX Course 2	XueTangX Course 3	ASSISTments Year 2004	ASSISTments Year 2005	ASSISTments Year 2006
Baselines	Raw	90.50	91.55	87.34	61.99	69.15	69.37	64.23	57.20	60.36
	LDA	76.83	75.04	75.17	66.93	71.39	70.44	62.79	66.50	66.44
	FM	93.98	94.21	88.89	65.31	70.50	69.43	74.10	69.32	70.20
	LadFG	-	-	-	68.35	72.63	73.56	-	-	-
	SDCDNTF2	94.42	96.55	93.43	67.23	71.50	71.27	61.52	59.79	60.94
	SDCDNTF4	94.37	96.47	93.46	67.15	72.56	71.19	61.49	62.32	60.88
	PairFac	94.44	95.54	93.19	67.40	71.96	72.22	69.54	67.59	70.26
Proposed Method	iDisc-1st	94.10	95.28	93.01	66.56	71.24	72.36	72.82	69.47	66.50
	iDisc-2nd	**95.37**	**96.86**	**94.58**	**69.39**	73.27	73.13	**78.37**	**72.84**	**73.26**
	iDisc-Comb.	94.79	96.14	93.76	69.35	**74.00**	**74.12**	77.47	72.65	71.49

5.3.3 Experiment Results. Table 4 shows the classification results. The raw features perform poorly, especially in XueTangX and ASSISTments dataset, with the score on accuracy in the range of 60-70%, which suggests the difficulty of the prediction task. LDA saw different performance, with noticeable drops in the edX dataset in comparison to raw features. We conjecture that the construction of the raw features results in a high dimensional and sparse feature space, which could potentially cause LDA to suffer from learning merely meaningful latent topics. Factorization machines slightly improve the performance. FMs can be considered as a generalization of tensor factorization with the additional modeling of interactions within each dimension [37]. Although we observe noticeable gains from FMs over Raw features and LDA, FMs do not perform as well as tensor-based methods. We suspect there might be two reasons for this: 1) the current feature space might not be as well tuned for general prediction tasks as it is for the more commonly seen recommendation tasks; 2) compared to tensor-based methods that only consider the interactions between different dimensions, FMs could potentially over-fit the interaction effects between and within dimensions in the training data. We observe that LadFG achieves large gains over the raw features for the XueTangX dataset. This indicates there could exist some hidden patterns that can capture the temporal elements of the behavior data. We also notice that tensor-based models such as PairFacand SDCDNTFperform better than LDA, especially in Course 2 and Course 3. This suggests that by systematically considering the multi-way interactions, the performance could be further improved. Finally, the best performance of iDisc is statistically comparable with the state-of-art LadFG and significantly better than the rest of the baselines. While LadFG is geared towards student performance predictions, iDisc can provide comparable prediction performance as well as meaningful patterns.

Summary. iDisc constructs students' embeddings that integrate relations between the multi-way interactions and the performance outcome. The quantitative experiment demonstrates the discriminative capability of iDisc, and iDisc outperforms the baselines in nine datasets from three MOOC platforms. Higher-level embeddings from iDisc have shown stronger discriminative powers over ones from the lower levels, which we will discuss in next section. Since there is no trivial solution in determing the rank of the tensor decomposition, we experimented with different rank settings for iDisc (e.g., {2, 4}, {3, 3}) and this observation still holds.

Table 5: Running time for varying number of observations.

#observations	10^2	10^3	10^4	10^5	10^6
running time/epoch	0.31s	0.35s	1.13s	1.23s	2.05s

5.4 Scalability

Since many of the education platforms have seen exponential growth in usage, scalable solutions of learning analytic are another critical aspect of adoption. In this section, we test the scalability of iDisc. In this experiment, we choose the ASSISTments dataset for the year of 2006, since it has the largest tensor settings in our experiment. We run iDisc with a varying number of entries in the data tensor, from $\{10^2, 10^3, 10^4, 10^5, 10^6\}$. Table 5 reports the average running time per epoch, and we observe that the running time scales almost linearly with the exponential increase in observations in the tensor. This result is consistent with the analysis in Section 4.1.4.

6 CONCLUSION

In this paper, we present a tensor-based learning framework, iDisc, to perform common and discriminative pattern discovery at multiple levels for understanding of high-dimensional student behavior and performance prediction in MOOCs. We first use tensors to represent each user's behavior, and construct coupled tensors to aggregate behavior for users with contrasting performance groups. Then, we iteratively identify the shared and distinct behavioral patterns at various levels, while revealing the hierarchical relationship between them to further increase the interpretability of the output. Finally, we use these patterns as anchors to generate the students' latent representation for down-stream performance prediction. Our qualitative examination of the patterns has shown the multi-level, multi-aspect and hierarchical characteristics of behavior patterns on the edX platform. The quantitative experiments, compared to both traditional predictive methods as well as existing discriminative tensor factorization models, suggest promising results by iDisc in several datasets from different MOOC platforms.

To the best of our knowledge, this is the first attempt to tackle the joint problem of discriminant tensor factorization and hierarchical pattern discovery for understanding such behavior on MOOC platforms. This enables the in-depth comprehension of students' multi-way behavior dynamics, as well as its association with course performance. Nevertheless, one of the limitations is that it merely

provides the relationships between the latent multi-way interaction and the performance outcome, with no intention to draw causal reasoning between them. In practice, iDisc can be developed as a plugin for MOOC platforms, where instructors can examine the multi-aspect contrasting behavior and connect the difference to the course outcome. Considering the XueTangX platform, one of the multi-aspect patterns could refer to a set of events at the beginning of the course that trigger from the server. if iDisc reveals its positive association to the success of students' course end performance, this pattern can be used as guidance of promotions for both the instructor and the platform to improve the students' learning outcome. Last, but not least, compared to other tensor factorization methods, iDisc provides a more efficient exploration of the multi-aspect patterns due to its multi-level nature. However, we understand that the interpretation of the multi-aspect pattern itself is not straightforward in general. In our future work, we would like to follow a more human-centric approach and develop a visual analytic system that helps domain experts interpret and understand the multi-aspect patterns.

ACKNOWLEDGEMENT

The authors would like to acknowledge the support from NSF #1634944, #1637067, and #1739413. Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of the funding sources.

REFERENCES

[1] Evrim Acar, Daniel M Dunlavy, and Tamara G Kolda. 2011. A scalable optimization approach for fitting canonical tensor decompositions. *Journal of Chemometrics* 25, 2 (2011), 67–86.

[2] Joon Hee Choi and S Vishwanathan. 2014. DFacTo: Distributed factorization of tensors. In *Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS'14*. 1296–1304.

[3] Andrzej Cichocki. 2014. Tensor networks for big data analytics and large-scale optimization problems. *arXiv preprint arXiv:1407.3124* (2014).

[4] Cody A Coleman, Daniel T Seaton, and Isaac Chuang. 2015. Probabilistic use cases: Discovering behavioral patterns for predicting certification. In *Proceedings of the Second ACM Conference on Learning at Scale*. ACM, 141–148.

[5] Christopher Donnelly. 2015. Enhancing Personalization Within ASSISTments. (2015).

[6] Michael Elad. 2010. From exact to approximate solutions. In *Sparse and Redundant Representations*. Springer, 79–109.

[7] Mi Fei and Dit-Yan Yeung. 2015. Temporal models for predicting student dropout in massive open online courses. In *Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 256–263.

[8] Xiao Fu, Kejun Huang, Evangelos E Papalexakis, Hyun-Ah Song, Partha Pratim Talukdar, Nicholas D Sidiropoulos, Christos Faloutsos, and Tom Mitchell. 2016. Efficient and distributed algorithms for large-scale generalized canonical correlations analysis. In *Proceedings of the 16th International Conference on Data Mining (ICDM)*. IEEE, 871–876.

[9] Nabeel Gillani, Rebecca Eynon, Michael Osborne, Isis Hjorth, and Stephen Roberts. 2014. Communication communities in MOOCs. *arXiv preprint arXiv:1403.4640* (2014).

[10] Thomas L Griffiths, Michael I Jordan, Joshua B Tenenbaum, and David M Blei. 2003. Hierarchical topic models and the nested chinese restaurant process. In *Proceedings of the 16th International Conference on Neural Information Processing Systems, NIPS'03*. 17–24.

[11] Julio Guerra, Shaghayegh Sahebi, Yu-Ru Lin, and Peter Brusilovsky. [n. d.]. The Problem Solving Genome: Analyzing Sequential Patterns of Student Work with Parameterized Exercises. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014*. 153–160.

[12] Sunil Kumar Gupta, Dinh Phung, Brett Adams, and Svetha Venkatesh. 2013. Regularized nonnegative shared subspace learning. *Data mining and knowledge discovery* 26, 1 (2013), 57–97.

[13] Sherif Halawa, Daniel Greene, and John Mitchell. 2014. Dropout prediction in MOOCs using learner activity features. In *Proceedings of the Second European MOOC Stakeholder Summit*. 58–65.

[14] John D Hansen and Justin Reich. 2015. Socioeconomic status and MOOC enrollment: enriching demographic information with external datasets. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*. ACM.

[15] R.A. Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. (1970).

[16] Neil T Heffernan and Cristina Lindquist Heffernan. 2014. The ASSISTments Ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education* 24, 4 (2014), 470–497.

[17] Roya Hosseini, Peter Brusilovsky, Michael Yudelson, and Arto Hellas. 2017. Stereotype modeling for Problem-Solving performance predictions in MOOCs and traditional courses. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 76–84.

[18] Chaoran Huang, Lina Yao, Xianzhi Wang, Boualem Benatallah, Shuai Zhang, and Manqing Dong. 2018. Expert recommendation via tensor factorization with regularizing hierarchical topical relationships. In *Proceedings of the International Conference on Service-Oriented Computing*. Springer, 373–387.

[19] Katy Jordan. 2014. Initial trends in enrolment and completion of massive open online courses. *The International Review of Research in Open and Distributed Learning* 15, 1 (2014).

[20] Rogier Kievit, Willem Eduard Frankenhuis, Lourens Waldorp, and Denny Borsboom. 2013. Simpson's paradox in psychological science: a practical guide. *Frontiers in psychology* 4 (2013), 513.

[21] Byung-Hak Kim, Ethan Vizitei, and Varun Ganapathi. 2018. GritNet 2: Real-Time Student Performance Prediction with Domain Adaptation. *arXiv preprint arXiv:1809.06686* (2018).

[22] Hannah Kim, Jaegul Choo, Jingu Kim, Chandan K Reddy, and Haesun Park. 2015. Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 567–576.

[23] Marius Kloft, Felix Stiehler, Zhilin Zheng, and Niels Pinkwart. 2014. Predicting MOOC dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*.

[24] Tamara Gibson Kolda. 2006. *Multilinear operators for higher-order decompositions*. Technical Report. Sandia National Laboratories.

[25] Stephan Lorenzen, Niklas Hjuler, and Stephen Alstrup. 2018. Tracking Behavioral Patterns among Students in an Online Educational System. In *Proceedings of the 11th International Conference on Educational Data Mining, EDM 2018, Buffalo, NY, USA, July 15-18, 2018*.

[26] Jorge Maldonado-Mahauad, Mar Pérez-Sanagustín, Pedro Manuel Moreno-Marcos, Carlos Alario-Hoyos, Pedro J Muñoz-Merino, and Carlos Delgado-Kloos. 2018. Predicting Learners' Success in a Self-paced MOOC Through Sequence Patterns of Self-regulated Learning. In *Proceedings of the European Conference on Technology-Enhanced Learning*. Springer, 355–369.

[27] Charalampos Mavroforakis, Isabel Valera, and Manuel Gomez-Rodriguez. 2017. Modeling the Dynamics of Learning Activity on the Web. In *Proceedings of the 26th International Conference on World Wide Web*. 1421–1430.

[28] Saurabh Nagrecha, John Z Dillon, and Nitesh V Chawla. 2017. MOOC dropout prediction: lessons learned from making pipelines interpretable. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 351–359.

[29] Ivan V Oseledets. 2009. Compact matrix form of the d-dimensional tensor decomposition. In *Proceedings of the International Symposium on Nonlinear Theory and its Applications*.

[30] Ivan V Oseledets and Eugene E Tyrtyshnikov. 2009. Breaking the curse of dimensionality, or how to use SVD in many dimensions. *SIAM Journal on Scientific Computing* 31, 5 (2009), 3744–3759.

[31] Alp Özdemir, Mark A Iwen, and Selin Aviyente. 2016. Multiscale tensor decomposition. In *Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 625–629.

[32] Alp Özdemir, Mark A Iwen, and Selin Aviyente. 2017. Multiscale Analysis for Higher-order Tensors. *arXiv preprint arXiv:1704.08578* (2017).

[33] Laura Pappano. 2012. The Year of the MOOC. *The New York Times* 2, 12 (2012).

[34] Judea Pearl et al. 2009. Causal inference in statistics: An overview. *Statistics surveys* 3 (2009), 96–146.

[35] Anh Huy Phan and Andrzej Cichocki. 2010. Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear theory and its applications, IEICE* 1, 1 (2010).

[36] Jiezhong Qiu, Jie Tang, Tracy Xiao Liu, Jie Gong, Chenhui Zhang, Qian Zhang, and Yufei Xue. 2016. Modeling and predicting learning behavior in MOOCs. In *Proceedings of the 9th ACM international conference on web search and data mining*.

[37] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the 10th International Conference on Data Mining (ICDM)*. IEEE, 995–1000.

[38] Giuseppe Ricci, Marco de Gemmis, and Giovanni Semeraro. 2014. Mathematical methods of tensor factorization applied to recommender systems. In *New Trends in Databases and Information Systems*. Springer, 383–388.

[39] Shaghayegh Sahebi, Yu-Ru Lin, and Peter Brusilovsky. 2016. Tensor factorization for student modeling and performance prediction in unstructured domain. In *Proceedings of the 9th International Conference on Educational Data Mining*.

[40] John Saint, Dragan Gašević, and Abelardo Pardo. 2018. Detecting Learning Strategies Through Process Mining. In *Proceedings of the European Conference on Technology-Enhanced Learning*. Springer, 385–398.
[41] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 650–658.
[42] Age Smilde, Rasmus Bro, and Paul Geladi. 2005. *Multi-way analysis: applications in the chemical sciences*. John Wiley & Sons.
[43] Panagiotis Symeonidis. 2016. Matrix and tensor decomposition in recommender systems. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 429–430.
[44] Feng Wang and Li Chen. 2016. A Nonlinear State Space Model for Identifying At-Risk Students in Open Online Courses. In *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*. 527–532.
[45] Wei Wang, Han Yu, and Chunyan Miao. 2017. Deep Model for Dropout Prediction in MOOCs. In *Proceedings of the 2nd International Conference on Crowd Science and Engineering*. ACM, 26–32.
[46] Xidao Wen, Yu-Ru Lin, and Konstantinos Pelechrinis. 2016. Pairfac: Event analytics through discriminant tensor factorization. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM.
[47] Xidao Wen, Yu-Ru Lin, and Konstantinos Pelechrinis. 2018. Event Analytics via Discriminant Tensor Factorization. *ACM Trans. Knowl. Discov. Data* (2018).
[48] Jibing Wu, Zhifei Wang, Yahui Wu, Lihua Liu, Su Deng, and Hongbin Huang. 2017. A Tensor CP decomposition method for clustering heterogeneous information networks via stochastic gradient descent algorithms. *Scientific Programming* 2017 (2017).
[49] Wanli Xing and Dongping Du. 2018. Dropout Prediction in MOOCs: Using Deep Learning for Personalized Intervention. *Journal of Educational Computing Research* (2018).
[50] Tiantian Zhang and Bo Yuan. 2016. Visualizing MOOC User Behaviors: A Case Study on XuetangX. In *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 89–98.