Large-Scale and Interpretable Collaborative Filtering for Educational Data

Kangwook Lee
KAIST
Daejeon, Korea
kw1jjang@kaist.ac.kr

Jichan Chung
KAIST
Daejeon, Korea
jichan3751@kaist.ac.kr

Changho Suh
KAIST
Daejeon, Korea
chsuh@kaist.ac.kr

ABSTRACT

Large-scale and interpretable educational data analysis is a key enabler of the next generation of education, and a variety of statistical models for such data and corresponding machine learning algorithms have been proposed in the literature. In this work, we introduce an interpretable multidimensional IRT model and propose an efficient algorithm that is highly scalable and parallelizable. Our approach provides improved human interpretability and greater scalability. We also provide experimental results on a real-world large-scale data set to demonstrate that our algorithm achieves as good prediction performance as the state of the arts. Further, leveraging the interpretability of our model, we offer an efficient and systematic method of identifying wrongly annotated tagging information.

KEYWORDS

Collaborative Filtering, Matrix Completion, Matrix Factorization, Recommendation System, Educational Data, Item Response Theory

1 INTRODUCTION

Educational data analysis aims at replacing the current old-fashioned education system with a fully personalized and automated education system [26]. Among many tasks of educational data analysis, personalized prediction of test responses based on the record of each individual learner is of the utmost importance. Another important task is question analysis, which necessitates human-interpretable statistical models of educational data. Thus, striking a critical balance between prediction performance and human interpretability is the key requirement that machine learning algorithms should satisfy. Moreover, the large amount of modern educational data, which is constantly being collected from online education platforms such as Massive Open Online Courses (MOOCs) [12], calls for the need of highly scalable learning algorithms that entail efficient time and space complexities.

A variety of statistical models of educational data such as test response data have been extensively studied in the literature. Among the proposed models, *item response theory (IRT)* models have received much attention due to their simplicity as well as superior human interpretability [20]. By identifying an implicit connection between these IRT models and the popular matrix completion framework, a few efficient algorithms have been recently developed [3, 19]. It is also demonstrated that their algorithms provide great prediction performances.

In this work, improving upon these algorithms, we propose a new machine learning algorithm for analyzing large-scale educational data. Our algorithm relies on a modified IRT model, and offers improved human interpretability and scalability, while providing as good prediction performance as the state-of-the-art algorithms. More specifically, our modified IRT model resolves the ambiguity issues of the original IRT models, which we will describe in Sec. 2.2 with greater detail. Further, our algorithm is highly scalable due to its efficient time complexity and space complexity as well as easily parallelizable. For comparison to the prior approaches, we collect a large-scale data set from an online education platform with more than 120000 users and more than 3800 multiple choice questions, each with 4 options. Using this data set, we demonstrate that the performance of our algorithm matches those of the state of the arts, while providing an improved human interpretability and greater scalability. Further, leveraging the interpretability of our model, we propose an efficient and systematic way of identifying wrongly annotated tagging information.

1.1 Notations

For a positive integer n, $[n] := \{1, 2, \cdots, n\}$. We shall use $\log(\cdot)$ to indicate the natural logarithm. Further, we denote by $[L_1; L_2; \cdots; L_n]$ the vertical concatenation of n row vectors $L_1, L_2, \ldots, L_n$.

2 RESPONSE MODEL AND ALGORITHM

2.1 Response Model

Item Response Theory (IRT) is a class of mathematical models of test responses [20]. While the unidimensional IRT models associate students and questions with scalar latent parameters, the multidimensional IRT (MIRT) models [25] associate them with multidimensional latent parameters, thus capturing the multiple factors affecting test responses, which we call *hidden concepts*.

Consider an education system with n students and m questions. The MIRT model assumes the following latent parameters associated with students and questions. For each i, $1 \le i \le n$, student i is associated with an r-dimensional row vector $L_i \in \mathbb{R}^{1\times r}$, where r denotes the upper bound on the number of *hidden concepts*. Similarly, for each j, $1 \le j \le m$, question j is associated with an r-dimensional row vector $R_j \in \mathbb{R}^{1\times r}$. In the MIRT model, it is assumed that the probability that student i correctly answers question j depends only on the inner product of L_i and R_j, i.e., $L_i R_j^T$. While this model is able to reflect multiple factors w.r.t. test responses as well as provides a reasonable level of human interpretability, the model is faced with inherent ambiguity issues. First, note that flipping signs of the k^{th} components of L_i and R_j does not alter the value of $L_i R_j^T$. Also, multiplying the k^{th} component of L_i by a constant factor and dividing the k^{th} component of R_j by the same factor yields the same value of $L_i R_j^T$. Hence, these incur ambiguities on interpretation of the latent parameters. For instance, $L_i(1) > L_i(2)$ does not necessarily imply that student i has a better understanding on the first hidden

concept relative to the second one. Similarly, $R_j(1) > R_j(2)$ does not necessarily imply that the first hidden concept is more important than the second one w.r.t. question j.

In order to resolve such ambiguities, we propose a simple variant of the MIRT model. In our model, we assume that student i is associated with an r-dimensional row vector L_i, of which each component is restricted to $[0, 1]$. Differently from the original MIRT model, our model allows for precise interpretation of the student features: the j^{th} component of L_i represents the level of student i's understanding on the j^{th} hidden concept. Similarly, question i is associated with an r-dimensional row vector $R_i \in [0, 1]^{1\times r}$, the components of which sum up to 1, i.e., $\sum_{j=1}^{r} R_i(j) = 1$. Here, the j^{th} component of R_i can be interpreted as the contribution of the j^{th} hidden concept to question i. We note that this modification fully resolves the ambiguity issue, enabling natural interpretation of the latent parameters. For instance, $L_i(1) > L_i(2)$ implies that student i has a better understanding on the first hidden concept than the second one. For notational simplicity, we define the student-concept matrix $L := [L_1; L_2; \cdots; L_n] \in [0, 1]^{n\times r}$ and the question-concept matrix $R := [R_1^T, R_2^T, \ldots, R_m^T] \in [0, 1]^{m\times r}$.

For student i and question j, the level of student i's understanding on question j is quantified as

$$X_{ij} = \sum_{k=1}^{r} L_i(k)R_j(k) = L_i R_j^T.$$

That is, the quantity is a weighted sum of the understanding levels of each hidden concept, where the weights are dictated by the composition of the question. Further, we assume a non-linear link function that maps the level of understanding to the probability of correct guess. More specifically, we assume the following logistic link function: Given X_{ij}, the probability that student i correctly answers question j is defined as

$$P_{ij} = \phi(X_{ij}) = \phi_a + \frac{1 - \phi_a}{1 + e^{-\phi_c(X_{ij}-\phi_b)}},$$

where ϕ_a, ϕ_b, and ϕ_c indicate some constants that are appropriately set, independently of questions and users. Here, ϕ_a, often called the guessing parameter, denotes the probability of correct guess when the level of understanding is zero, and (ϕ_b, ϕ_c) are introduced for a proper normalization[1]. Let $X := [X_{ij}] \in [0, 1]^{n\times m}$ denote the understanding level matrix and $P := [P_{ij}] \in [0, 1]^{n\times m}$ denote a matrix w.r.t. the probability of correct-answer. Note that $X = LR^T$ and $P = \phi(X)$, where $\phi(\cdot)$ is applied component-wise. Finally, we assume that $Y_{ij} \in \{0, 1\}$, which represents whether student i guessed the correct answer for question j ($Y_{ij} = 1$) or not ($Y_{ij} = 0$), follows a Bernoulli random distribution with probability P_{ij}. We denote by Ω the set of student-question pairs for observed responses. Further, we denote by $\Omega_{i\star}$ and $\Omega_{\star j}$ the set of question indices attempted by student i and the set of indices of users who attempted question j, respectively. That is, $\Omega_{i\star} = \{j|(i, j) \in \Omega\}$, and $\Omega_{\star j} = \{i|(i, j) \in \Omega\}$.

We note that the model described above cannot capture the inherent difficulties of problems. In order to resolve this issue, we introduce the following two auxiliary concepts: the $(r + 1)^{\text{th}}$ concept is the one that no one knows, and the $(r + 2)^{\text{th}}$ concept is the one that everyone knows. This can be imposed by setting $L_i(r + 1) = 0$ and $L_i(r+2) = 1$ for all i. On the other hand, the parameters $R_i(r+1)$ and $R_i(r + 2)$ for all i are treated equally as the other hidden parameters, and hence need to be estimated. In order to see how these auxiliary concepts help model atypical questions, consider an extreme case. Imagine that question j consists of a concept that is not known to everyone. One would like to model this case in a way that every user, regardless of their backgrounds, will randomly guess the answer of the question. This situation can be easily captured under our model by setting $R_j(r + 1) = 1$. As $L_i(r + 1) = 0$ for all i, the understanding level of student i on question j is zero, i.e., $L_i R_j^T = X_{ij} = 0$, implying $P_{ij} = \phi_a$. Thus, this immediately models the situation where all the users will randomly guess the answer of question j.

In this variant of MIRT model[2], the goal of educational data analysis is to estimate L and R from the observed test responses $(Y_{ij})_\Omega$, and to predict missing components of Y from the estimated latent parameters.

2.2 Algorithm

We begin with a brief review of the matrix completion problem since it is intimately related to the inference problem of our interest.

The goal of matrix completion (also known as matrix factorization) is to fill a low-rank matrix with partially revealed entires [2, 8–10]. It has been shown powerful for tackling various collaborative filtering problems such as the recommendation problem. In [10], Candes and Rao show that, under some mild conditions, one can reliably fill a square matrix of size n by n and of rank r if the number of observed entries is the order of $nr\text{polylog}(n)$ by solving an optimization problem called the nuclear norm minimization problem. A similar result holds even when the observed entries are not exact but noisy. In addition to such convex program approaches, many other efficient algorithms (e.g., spectral methods, non-convex algorithms, stochastic algorithms) have been proposed in the literature [7, 16, 24]. The 1-bit low-rank matrix completion [13] is a variant of the original problem. The goal of the problem is to fill a *binary*-valued matrix assuming that the observed entries are realization of Bernoulli random variables whose probabilities being 1 are governed by a hidden low-rank matrix.

The 1-bit low-rank matrix completion problem has a strong connection to the inference of latent parameters under the MIRT model. One can view the observed test responses as a binary-valued matrix with missing entries. This way, the inference of the latent parameters L and R reduces to a variant of the 1-bit matrix completion. In [3], Berner et al. first observe such connection. In [19], Lan et al. introduce a variant of the MIRT model, which imposes sparsity constraints on R_j's for improved interpretability, and propose an algorithm that can efficiently estimate the latent parameters. The dealbreaker model, proposed in [17], is a nonlinear latent variable model under which the student's success probability depends only on his/her weakest concept understanding. The authors also propose an algorithm based on the alternating direction method of multipliers (ADMM) framework [5].

[1] We remark that though seemingly similar, ϕ_b is irrelevant to the easiness parameters of the classic IRT models.

[2] We remark that our model with these auxiliary concepts can also be seen as an alternative form of the M3PL latent trait model since it captures multidimensional item discrimination, item difficulty, as well as different guessing probability for each problem.

Algorithm 1

Input: observed responses $(Y_{ij})_{(i,j)\in\Omega}$, index set Ω
Initialize $L^{(0)}$ and $R^{(0)}$ uniformly at random in $[0, 1]$.
Normalize every $R_i^{(0)}$ such that $\sum_{j=1}^{r} R_i^{(0)}(j) = 1$.
$k = 0$
repeat
  $L = L^{(k)}$, $R = R^{(k)}$
  Shuffle Ω
  for (i, j) **in** Ω **do**
    $\beta_L = 1 - \frac{\mu\alpha_k}{|\Omega_{i\star}|}, \beta_R = 1 - \frac{\mu\alpha_k}{|\Omega_{\star j}|}$
    $\beta = \frac{\phi_c\left(Y_{ij} - \phi(L_i R_j^T)\right)}{\phi(L_i R_j^T)(1+e^{-\phi_c(L_i R_j^T - \phi_b)})}$
    $L_i = \Pi_{P_L}\left(\beta_L L_i + \beta R_j\right)$
    $R_j = \Pi_{P_R}\left(\beta_R R_j + \beta L_i\right)$
  end for
  $k = k + 1$
  $L^{(k)} = L$, $R^{(k)} = R$
until convergence
Output: $L^{(k)}$ and $R^{(k)}$

While the prior algorithms have shown superior prediction performances, most of them can hardly scale and/or can be hardly parallelized. In order to design a scalable and parallelizable algorithm for a large-scale educational data analysis, we develop an algorithm based on the projected Stochastic Gradient Descent (SGD) method, which can be readily deployed on a parallel/distributed computing platform as shown in [15, 24].

We now present our algorithm. Our algorithm is highly scalable since its time and space complexities are linear in the number of observed test responses. Further, it is inherently online: an already trained model can be efficiently retrained when new test responses are revealed.

Our algorithm attempts to find the maximum likelihood (ML) estimator of X given a set of observation $[Y_{ij}]_{(i,j)\in\Omega}$. Equivalently, the ML estimator can be found by solving a minimization problem whose objective function is the negative of the log likelihood of the observed entries. In order to encourage X to reflect a low-rank structure, we also add to the objective function the nuclear norm regularization term [10]. That is, we formulate an optimization problem as:

$$\begin{aligned} \min_{L,R} \quad & \sum_{(i,j)\in\Omega} \ell(Y_{ij}, P_{ij}) + \mu\|LR^T\|_* \\ \text{s.t.} \quad & 0 \le L_{ij} \le 1\,,\ 0 \le R_{ij} \le 1\,,\ \forall i, j, \\ & P = \phi(LR^T),\ \sum_j L_{ij} = 1\,,\ \forall i\,, \end{aligned} \tag{P1}$$

where $\ell(Y_{ij}, P_{ij})$ indicates the negative log-likelihood of the observed response Y_{ij} when $P(Y_{ij} = 1) = P_{ij}$:

$$\ell(Y_{ij}, P_{ij}) = -Y_{ij}\log(P_{ij}) - (1 - Y_{ij})\log(1 - P_{ij}).$$

We intend to approximate the optimization problem (P1) with (P2) (see below) by replacing the nuclear norm of LR^T with the sum of the squared Frobenius norms of L and R. This approximation is based on the following property [23]: the nuclear norm of a matrix X is equal to the minimum sum of the squared Frobenius norms of L and R such that $X = LR^T$.

$$\begin{aligned} \min_{L,R} \quad & \sum_{(i,j)\in\Omega} \ell(Y_{ij}, P_{ij}) + \frac{\mu}{2}\left(\|L\|_F^2 + \|R\|_F^2\right) \\ \text{s.t.} \quad & 0 \le L_{ij} \le 1\,,\ 0 \le R_{ij} \le 1\,,\ \forall i, j \\ & P = \phi(LR^T),\ \sum_j L_{ij} = 1\,,\ \forall i\,. \end{aligned} \tag{P2}$$

Indeed, any local minimum of (P2) is known to match that of the global minimum of the original problem (P1) under mild conditions, and this agreement can be further certified by checking rank deficiency of L and R [24].

As a specific choice of the algorithm for solving (P2), we make use of the projected Stochastic Gradient Descent (SGD) method. A formal description of our algorithm is given in Algorithm 1. The algorithm starts with randomly initialized $L^{(0)}$ and $R^{(0)}$, and then iteratively updates sequences of $L^{(k)}$ and $R^{(k)}$ as follows. At the beginning of each epoch, we randomly shuffle the index set Ω. For each pair of indices from Ω, say (i, j), we update L_i and R_j as in Algorithm 1 where $\Pi_{P_L}(\cdot)$ and $\Pi_{P_R}(\cdot)$ are projections of a vector onto the spaces of feasible L's and R's, respectively. This procedure is repeated until $L^{(k)}$ and $R^{(k)}$ converge. Note that each epoch's runtime consists of time to shuffle data points and time to update parameters $|\Omega|$ times. Since time to update parameter takes $O(r)$, the total time complexity, if a constant number of epochs is run, is $O(r|\Omega|)$.

Note that the projected SGD is known to converge to a globally optimal solution when the objective function and the regularization terms (including those induced by constraints) are convex [21]. The objective function of (P2), however, is non-convex due to the product term LR^T, so we run the above algorithm multiple times with different initialization points.

A simple variation of our algorithm is the one intended for the case when both R and the test response data set Y are given. It can be shown that given R and observed responses Y_Ω, the optimization problem (P2) reduces to a set of independent logistic regression problems, each being with linear constraints. Hence, one can estimate L by solving all of the logistic regression problems, and concatenating the estimated user features L_i's.

2.3 Comparison with Existing Algorithms

In this section, we compare our proposed algorithms with the existing algorithms in the literature. See Table 1 for the summary.

We first compare the algorithms that can estimate latent variables. As a specific instance of the IRT model, consider the two-parameter logistic model (2PL) [20]. Assume that a gradient descent method is applied to solve the corresponding ML estimation problem. The 2PL model assumes that the probability of correct guess depends on the sum of the user latent variable and the question latent variable. While it allows for less ambiguous interpretation of estimation results, it may not be able to capture a complex structure due to its limited model complexity. On the other hand, the original MIRT model [25] has the ability to express more complex models but fails to provide consistent and interpretable estimation results. SPARFA [18], a variant of the MIRT model, has much improved human interpretability

Model & Algorithm	Class	Human-Interpretable	Requires R	Scalable & Parallelizable	Online
2PL [20]	IRT, Affine	✓	✗	✗	✗
MIRT [25]	MIRT, Affine	✗	✗	✗	✗
SPARFA [19]	MIRT, Affine	✓✓	✗	✗	✓
Dealbreaker [17]	Nonlinear	✓✓	✗	✓	✗
Ours (Sec. 2.2)	MIRT, Affine	✓	✗	✓	✓
G-DINA [14]	Nonlinear	✓✓	✓	✗	✗
Ours with R (Sec. 2.2)	MIRT, Affine	✓	✓	✓	✓

Table 1: Machine learning algorithms for educational data analysis

Name	n	m	$\|\Omega\|$	$\|\Omega\|/(nm)$
Full	123973	3835	8861570	1.86%
Filtered	16065	1999	2983327	9.29%

Table 2: Data sets

due to the sparse nature of their estimated parameters. More precisely, by imposing sparsity constraints on R, their algorithm is able to identify a few most important hidden concepts associated with each question. Further, it scales well to high-dimensional problems since it relies on a first-order method called the FISTA framework [1]. However, it is not clear whether it can be easily parallelized. On the other hand, the dealbreaker model [17] is based on the ADMM framework [5], and hence can be easily parallelized.

Recall that when the precise estimate of R is provided, our algorithm reduces to multiple instances of convex problems, each of which resembles logistic regression. This is because when R is fixed, the objective function and the constraints of (P2) can be decomposed into n instances of a simple logistic regression. The G-DINA (generalized deterministic inputs, noisy "and" gate) model [14] can be deployed when such question tagging information is available, and allows for highly interpretable results. However, unlike our algorithm, the existing algorithms for the G-DINA model are neither scalable nor parallelizable.

3 EXPERIMENTAL SETUP AND RESULTS

3.1 Data Set

We first collected a pool of *TOEIC* (Test Of English for International Communication) questions. TOEIC is a test of English for international communication, and each test is composed of 150 multiple-choice questions with 4 options each. We first created the question pool of 3835 TOEIC questions. With this question pool, we have collected a large response data set via an online TOEIC education platform. From 1/1/2016 to 1/15/2017, a total of 123973 students had signed up for the platform, and a total of 8861570 responses had been collected. Note the extremely low density of the response matrix, which amounts to about 1.86%.

In order to obtain a high quality data set, we preprocess the raw data set as follows. We first removed the students who had attempted less than 30 questions during the observation period or had spent less than 3 seconds for more than or equal to 95% of their attempts. Similarly, we filtered out students whose correct answer rate is less than or equal to 30%[3] After we obtained the refined set of students, we filter out the questions that are responded less than 400 distinct students.

With the aforementioned filtering process, we obtained the filtered data set consisting of $|\Omega| = 2983327$ responses of $n = 16065$ students on $m = 1999$ questions. Note that the density of the observation matrix is about 9.29%. The size of the original data set and the filtered data set are summarized in Table 2.

3.2 Algorithm Implementation and Specification

We randomly divide the filtered data set into the training set (90%) and the test set (10%). All the experiment results reported in this section are with respect to the test set. We then conduct a heuristic optimization for finding the optimal hyper-parameters such as the regularization parameter μ, the sequence of step sizes, and etc. As a result, we chose $\mu = 1$; the step size is initialized as $\alpha_0 = 0.1$, and is decreased by a multiplicative factor of $10^{0.3}$ whenever the validation score stops improving for 3 epochs in a row. For the link function $\phi(\cdot)$, we use $\phi_a = 0.25$, $\phi_b = 0.5$ and $\phi_c = 10$. The rationale behind these choices is that one can correctly guess the answer of a question without knowing anything about a 4-choice question with probability at least 0.25.

We implement our algorithm in Python. In addition to our approach, we also evaluate the prediction performances of some of the approaches described in Sec. 2.3: we fit our data set to the 2PL model [20] using `mirt` R package [11], and to the vanilla MIRT model [25] using the 1-bit matrix completion algorithm of [13].

3.3 Prediction Performance

In this section, we evaluate the prediction performances of various algorithms. More precisely, we run various algorithms with the training set and measure the prediction (classification) performances. A prediction outcome for an unobserved test response is called a true positive (negative) if the predictor correctly guessed that the student will respond to the question with a correct (wrong) answer. Similarly, a prediction outcome is a false positive (negative) if the predictor made a wrong guess that the student will respond to the question with a correct (wrong) answer. We denote the number of true positives, false positives, true negatives, and false negatives by tp, fp, tn, and fn, respectively. For the performance metric, we consider the area

[3]The rationale behind these filtering conditions is that students not satisfying these conditions are likely to be ones who simply wanted to try out and explore the mobile applications for fun.

ALGORITHM	AUC	NLL
M2PL	**0.7775**	**0.5209**
MIRT (L,$r = 2$)	0.7674	0.5413
MIRT (L,$r = 4$)	0.7695	0.5361
MIRT (P,$r = 2$)	0.7696	0.5338
MIRT (P,$r = 4$)	0.7692	0.5320
OURS ($r = 2$)	**0.7760**	**0.5223**
OURS ($r = 4$)	0.7707	0.5277

Table 3: Prediction performances on the filtered data set. AUC denotes the area under curve (AUC) of a receiver operating characteristic (ROC) curve, and NLL denotes the negative of log likelihood. For the MIRT model, we test both logistic link functions (denoted by 'L') and probit link function (denoted by 'P'). For the MIRT model and our algorithm, we vary the number of hidden concepts $r \in \{2, 4\}$.

under curve (AUC) of a receiver operating characteristic (ROC) curve. For a classification threshold $\theta \in [0, 1]$, the ROC curve is a collection of pairs $(\text{fpr}(\theta), \text{tpr}(\theta))$. Note that the ROC curve of a random predictor is a line segment connecting $(0, 0)$ and $(1, 1)$, and that of a perfect predictor is line segments connecting $(0, 0)$, $(0, 1)$, and $(1, 1)$. Thus, the area under curve (AUC) of a ROC curve can represent the classification performance of a predictor: the larger the AUC is, the better the prediction performance is. We also measure the negative of log likelihood (NLL) of each prediction algorithm.

The prediction performances of various learning algorithms are summarized in Table 3. For the MIRT model, we test both logistic link function (denoted by 'L') and probit link function (denoted by 'P'). For the MIRT model and our algorithm, we evaluate the prediction performances with $r \in \{2, 4\}$. We observe that when we set $r > 4$, the prediction performance is strictly worse[4]. For each configuration, the AUC and NLL are measured 20 times with randomly divided data sets, and the average values are shown. We can see that the best prediction performance is achieved by the M2PL model, and our algorithm closely matches the best performance.

Thanks to the scalability of our algorithm, we could also measure the AUC performance of our algorithm w.r.t. the full data set: the average AUC is observed to be 0.7778.

3.4 Tag Correction via Interpretable Results

In a large-scale education system, questions are associated with 'concept tags', and such associations are usually judged by experts in a manual way. However, such tagging information is prone to errors due to human errors, inherent ambiguity, atypical questions, etc. Thus, identifying those wrongly tagged questions and correcting them are a key to maintain high-quality tagging information, which is crucial for providing suitable learning materials in a personalized education system.

We first explain how the improved interpretability of our model allows for an efficient tag correction procedure. First, if there exist the sign ambiguity and the scale ambiguity, the question features of similar questions are not necessarily close to each other. Secondly, question features obtained under ambiguous models are not invariant across different training instances. For example, when one has to rerun the training algorithm from scratch for some reasons (such as new dataset arrival, batch model update, algorithm modification, etc.), all question features change, and such a tagging correction procedure has to be restarted from scratch.

Leveraging the improved interpretability of our algorithm, we now present a systematic way of identifying wrongly tagged questions. Consider the case of $r = 2$. Under our model, the quantity $\gamma = R_i(1)/(R_i(1) + R_i(2))$ can be interpreted as the relative fraction of hidden concept 1 w.r.t. question i. If our model can well explain the data set, questions of the same type are supposed to have similar values of γ. Thus, by inspecting the values of γ of the questions annotated with the same tag, one may be able to detect wrongly tagged questions.

In order to conduct the experiment, we had a small subset of the TOEIC question pool tagged by experts as follows. The 15 hired experts first investigated the question pool, and then came up with a set of 69 tags, which were considered useful and necessary for describing the questions in the question pool. More specifically, each question was randomly assigned to at least two experts, and the experts tagged each of the assigned questions with the most relevant concept. We develop an online tagging system where the experts were able to individually work on the assigned questions. In order to reduce systematic bias between experts, we revealed the first reviewer's response to a question to the second reviewer of the question so that the second reviewer can adjust the response of the first reviewer[5].

After every question is tagged, we compare the values of γ of the questions of the same tag, and identified a large number of wrongly tagged questions, of which a few instances are shown below. We plot in Fig. 1 the values of γ of all the questions tagged with the tag '`for oneself/by oneself/on one's own`'. There are 12 questions, denoted by Qi for $i \in [12]$, which are identified as questions that require the understanding on the usage of 'for oneself', 'by oneself', or 'on one's own'. These questions are for testing whether students can fill a blank in a sentence with a grammatically correct word or phrase. Each question is tagged with the key phrase that one needs to fully understand in order to correctly answer the question. However, we can observe 3 clusters of the questions: the first cluster consisting of Q1 and Q2, the second one in the middle, and the third one consisting of Q9 to Q12.

It turns out that the questions in the first and third clusters are associated with incorrect tags. Table. 4 shows two correctly tagged questions and two incorrectly tagged ones. It is clear that the first two questions are about 'for oneself' but the other two questions are irrelevant to the tag. For instance, Q10 is asking whether students can fill the missing pronoun 'them'. We conjecture that this problem is wrongly tagged because the problem may seem relevant to the concept `on one's own`. Similarly, Q11 is clearly wrongly tagged since it is about whether students can fill the missing blank with a correct reflexive pronoun.

[4]Note that choosing a larger value for r, the upper bound on the true rank, does not necessarily improve the prediction performance due to overfitting.

[5]We could not measure inter-rater reliability (IRR) since the responses of different experts were dependent under our scheme.

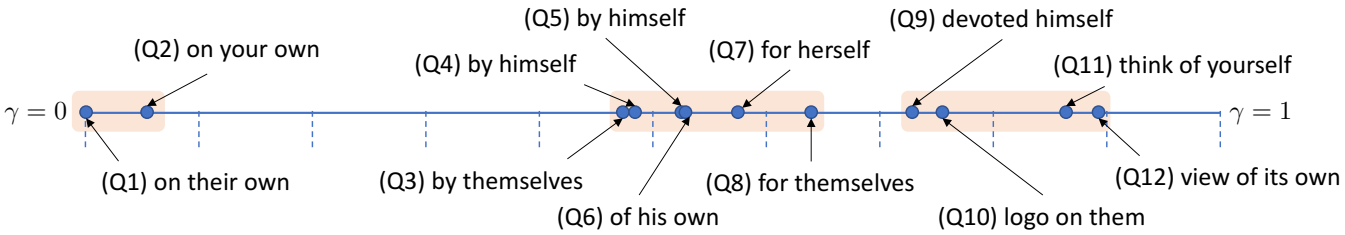
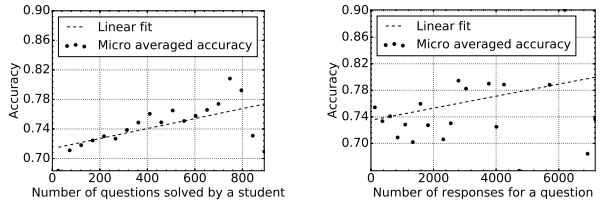


Figure 1: γ values of the questions tagged with '`for oneself/by oneself/on one's own`'

Q7. MS. GRANT HAS ASKED FOR A MINI-BUS FROM THE AIRPORT TO THE CLIENT'S OFFICES FOR **(HERSELF)** AND THE ENTIRE SALES TEAM THIS MONDAY.	**Q8.** THE MANUFACTURING PLANT'S POLICY ALLOWS WORKERS TO SPEAK FOR **(THEMSELVES)** IN CASES WHERE COMPENSATION IS SOUGHT FOR AN INJURY SUFFERED ON THE JOB.	**Q10.** THE COMPANY PULLED THEIR T-SHIRTS OFF FROM STORE SHELVES WHEN THE PUBLIC COMPLAINED ABOUT THE OFFENSIVE LOGO ON **(THEM)**.	**Q11.** IT'S NEVER A GOOD IDEA TO THINK OF **(YOURSELF)** AS SUPERIOR TO YOUR SUPERIORS, OR YOU JUST MIGHT GET FIRED.

Table 4: A few questions tagged with '`for oneself/by oneself/on one's own`'



(a) Prediction accuracy as a function of the number of questions solved by a user

(b) Prediction accuracy as a function of the number of responses for a question

Figure 2: Prediction performance as a function of the number of per-student, per-question observations.

4 CONCLUSION

In this work, we proposed a new algorithm based on a human-interpretable MIRT model, which is both scalable and parallelizable. Using the large data set collected from an online education platform, we observe that our algorithm can achieve as good prediction performance as the state of the arts. Moreover, we show that the improved interpretability of our model allows for an efficient and systematic way of identifying wrongly tagged questions. We conclude the paper by discussing a few interesting open problems and some aspects of our current model that are subject to improvements.

4.1 Correlation Between the Number of Responses and Prediction Accuracy

In our data set, the number of responses per student and the number of responses per question widely vary. One important question is: how the prediction performance changes when the number of questions per student (or per question) increases. To answer this question, we first bin the students according to the number of responses per student, and then micro-average the prediction accuracies of the students of the same bin. Similarly, we bin the questions according to the number of responses per question, and then micro-average the prediction accuracies. Plotted in Fig. 2 are micro-averaged accuracy as a function of the number of questions attempted by students and that as a function of the number of responses per question. We use the bin size of 50 for Fig. 2(a), and the bin size of 250 for Fig. 2(b). From Fig. 2(a), we can observe that the predicted accuracy of a student's responses linearly increases as the number of questions submitted by the student increases. Similarly, the predicted accuracy of the responses for a question linearly increases with the number of responses for the question. Thus, one can determine using this procedure the number of responses per student or per question with which a personalized education system can provide prediction services with high enough accuracy.

4.2 Incorporation of Other Forms of Data

While we used the binary response data only, the actual response data set contains several additional sources of side information such as the option chosen by students, the options marked wrong by students, the time taken to respond to a question, and etc. By incorporating the other forms of data with a more complicated model, one may be able to obtain better estimates of students and questions, and hence to provide superior prediction performance as well as personalized learning of a better quality. For instance, the nominal response model (NRM) proposed in [4] can model the probability of students responding to a certain option of a question. In [22], Ning et al. propose a new model for option responses with human-interpretable outputs, and show that the new model fits better with real world data as well. It is an interesting future direction to study how one can apply a similar collaborative filtering-approach under such models capturing option responses. In [6], Brinton et al. show that one can predict students' future performances on quizzes using video-watching clickstream data from MOOCs. It is an interesting open question whether a unified model that uses both response data and video-watching clickstream data can achieve higher prediction performance.

4.3 Time-varying L

Our response model implicitly assumes that the level of students' understanding is *time-invariant*. If the data set is collected over a long time period during which student's level of understanding is likely to fluctuate, such an assumption may totally fail, and the estimated L will be close to the time average of L, which is less informative for predicting future responses.

If one is given with an enormous amount of data, there is a simple fix: one can simply keep fresh responses collected over a short time period only: a time-invariant model for students' understanding will fit better for a shorter range of time. The number of responses in the dataset, however, decreases when one reduces the data collection period, possibly deteriorating the prediction performance.

In order to resolve this issue, in [18], Lan et al. have proposed a time-variant model for learning analytics capturing the time varying levels of understanding of learners. We believe that such a time-variant model can take advantage of a large amount of data without compromising the fitness of the model.

4.4 Sparsity of R

While it is reasonable to believe that among many concepts only a few are required to correctly answer a question, we observe that our collaborative filtering algorithm usually results in a dense question-concept matrix R. Therefore, imposing sparsity on R can potentially allow for a better model and hence an improved prediction performance. In [19], Lan et al. propose a collaborative filtering that can find a sparse question-concept matrix R by incorporating the ℓ_1 regularization term into the objective function of the optimization problem. The authors observe a superior prediction performance of their proposed sparse model compared with the non-sparse model proposed in [3]. Inspired by this observation, we also measured the performance of the variation of our algorithm where the ℓ_1 regularization term is incorporated but we did not observe an improvement in prediction performance with our data set. Even though we could not observe an improvement in prediction performance with our data set, we believe that the sparse models, capturing the natural sparsity of R, will result in more accurate estimates in general.

4.5 Mixture Models and Outlier Detection

All the models described in this paper assume a common assumption: one model fits all the students and questions. This could be the case for small-scale educational data such as those collected from classrooms but not for data collected from a large-scale education platform with hundreds of thousands of students with completely different backgrounds. For instance, in an online education system where students freely choose questions to work on and do not get penalized for guessing wrong answers, some students might recklessly solve questions, resulting in random responses, which do not conform existing response models. Similarly, some questions in a large question pool may not conform the typical pattern of the other questions. Hence, an accurate mixture model capturing such outliers can greatly enhance the performance prediction.

REFERENCES

[1] Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* 2, 1 (2009), 183–202.

[2] Robert Bell, Yehuda Koren, and Chris Volinsky. 2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 95–104.

[3] Yoav Bergner, Stefan Droschler, Gerd Kortemeyer, Saif Rayyan, Daniel Seaton, and David E Pritchard. 2012. Model-Based Collaborative Filtering Analysis of Student Response Data: Machine-Learning Item Response Theory. *International Educational Data Mining Society* (2012).

[4] R Darrell Bock. 1972. Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika* 37, 1 (1972), 29–51.

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3, 1 (2011), 1–122.

[6] Christopher G Brinton, Swapna Buccapatnam, Mung Chiang, and H Vincent Poor. 2016. Mining MOOC Clickstreams: Video-Watching Behavior vs. In-Video Quiz Performance. *IEEE Transactions on Signal Processing* 64, 14 (2016), 3677–3692.

[7] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20, 4 (2010), 1956–1982.

[8] Emmanuel J Candes and Yaniv Plan. 2010. Matrix completion with noise. *Proc. IEEE* 98, 6 (2010), 925–936.

[9] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9, 6 (2009), 717–772.

[10] Emmanuel J Candès and Terence Tao. 2010. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory* 56, 5 (2010), 2053–2080.

[11] R Philip Chalmers and others. 2012. mirt: A multidimensional item response theory package for the R environment. (2012).

[12] John Daniel. 2012. Making sense of MOOCs: Musings in a maze of myth, paradox and possibility. *Journal of interactive Media in education* 2012, 3 (2012).

[13] Mark A Davenport, Yaniv Plan, Ewout van den Berg, and Mary Wootters. 2014. 1-bit matrix completion. *Information and Inference* 3, 3 (2014), 189–223.

[14] Jimmy De La Torre. 2011. The Generalized DINA Model Framework. *Psychometrika* 76, 2 (2011), 179–199. https://doi.org/10.1007/s11336-011-9207-7

[15] Rainer Gemulla, Erik Nijkamp, Peter J Haas, and Yannis Sismanis. 2011. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 69–77.

[16] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. 2013. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM, 665–674.

[17] Andrew Lan, Tom Goldstein, Richard Baraniuk, and Christoph Studer. 2016. Dealbreaker: A Nonlinear Latent Variable Model for Educational Data. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48 (ICML'16)*. JMLR.org, 266–275. http://dl.acm.org/citation.cfm?id=3045390.3045420

[18] Andrew S Lan, Christoph Studer, and Richard G Baraniuk. 2014. Time-varying learning and content analytics via sparse factor analysis. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 452–461.

[19] Andrew S Lan, Andrew E Waters, Christoph Studer, and Richard G Baraniuk. 2014. Sparse factor analysis for learning and content analytics. *Journal of Machine Learning Research* 15, 1 (2014), 1959–2008.

[20] Frederic M Lord. 1980. *Applications of item response theory to practical testing problems*. Routledge.

[21] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization* 19, 4 (2009), 1574–1609.

[22] Ryan Ning, Andrew E Waters, Christoph Studer, and Richard G Baraniuk. 2015. SPRITE: A Response Model For Multiple Choice Testing. *arXiv preprint arXiv:1501.02844* (2015).

[23] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review* 52, 3 (2010), 471–501.

[24] Benjamin Recht and Christopher Ré. 2013. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation* 5, 2 (2013), 201–226.

[25] Mark Reckase. 2009. *Multidimensional item response theory*. Vol. 150. Springer.

[26] George Siemens. 2010. What are learning analytics? http://www.elearnspace.org/blog/2010/08/25/what-are-learning-analytics/. (2010). Accessed: 2016-09-30.