Priyanshu Arora, Aidan Brasseur, Sky Li

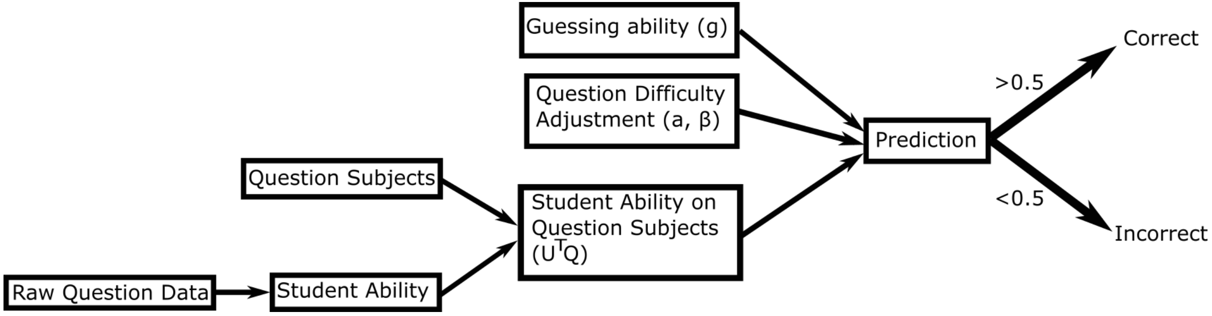
We want to predict if a student can correctly answer a specific question based on the student's previous answers to other questions and other students' responses. In this dataset, we are given a set of students along with all the questions each student answered marked as correct or incorrect. We are also given the subject areas each question covers. There are 388 total subject areas. Please refer to the Item Response Theory model linked at the end of this paper to understand the base model we extended.

Overview

We have extended the IRT model with several additions. $\mathbf{U}_i$ is a 388×1 vector that represents the i-th student's ability in each of 388 subjects. $\mathbf{Q}_j$ is a constant 388×1 vector of 1s and 0s, where a 1 in the k'th index represents that the question is associated with the k'th subject. Using these two parameters, then $\mathbf{U}_i^T \mathbf{Q}_j$ represents the sum of the abilities of the student i in the subjects that are relevant to the question j. β_j remains the same, representing the difficulty of the question j. Inspired by the work of Luca Benedetto, we added the parameters a and g. a_j represents the rate at which the probability of answering the j-th question correctly changes with respect to the ability of the student. In other words, a_j adjusts the steepness of the IRT function. This helps to capture the possibility that certain questions can have a difficulty that doesn't exactly scale with student ability. For example, if a question requires trigonometry knowledge that not every student has, but is extraordinarily easy for those students that do, a_j will allow the model to scale the IRT function for that question so that that nuance is properly captured. g represents the probability that a student without any ability in the relevant subjects will be able to correctly guess the answer to the question. With these additional parameters, we hope that the model will be able to better capture the nuances of the data set, thus reducing its overall bias. After these modifications, our model becomes:

$$p(c_{ij} = 1 | \mathbf{U}_i, \beta_j, \mathbf{Q}_j, a_j, g) = g + \frac{(1-g)\exp(a_j(\mathbf{U}_i^T \mathbf{Q}_j - \beta_j))}{1 + \exp(a_j(\mathbf{U}_i^T \mathbf{Q}_j - \beta_j))}$$

This model can be visualized as follows:



To initialize the matrix $\mathbf{Q}$, we utilized the question metadata found in question_meta.csv, parsing the list of subjects for each question j and constructing the row $\mathbf{Q}_j$ accordingly. As each question has 4 possible answers (A, B, C, D), we decided to assign g to be the constant value of 0.25. As the students ability in the corresponding subjects of question j ($\mathbf{U}_i^T \mathbf{Q}_j$) approaches $-\infty$, their probability of answering the question correctly approaches 0.25, which makes sense as a student will always have a 1 in 4 chance of guessing correctly even with no knowledge.

We expected this model to perform better in particular because we felt that expanding the model in this way would capture patterns within a student's areas of expertise in certain subjects rather than simply considering their overall ability.

One thing to note about the data set is that every question was categorized by subject 0, Maths. This means that the first entry of $\mathbf{U}_i$ can also be interpreted as a general ability in maths similar to the role that θ_i was playing in the original model. To illustrate this relationship, we will provide the following example:
Assume that there are 3 subjects: math, calculus, and linear algebra with ids 0, 1, 2 respectively.
Assume that student i has only answered calculus questions, but has answered these very strongly.

An example of what our $\mathbf{U}_i$ vector could look like is:

$$U_i = \begin{bmatrix} 1.5 \\ 1.5 \\ 0 \end{bmatrix}$$

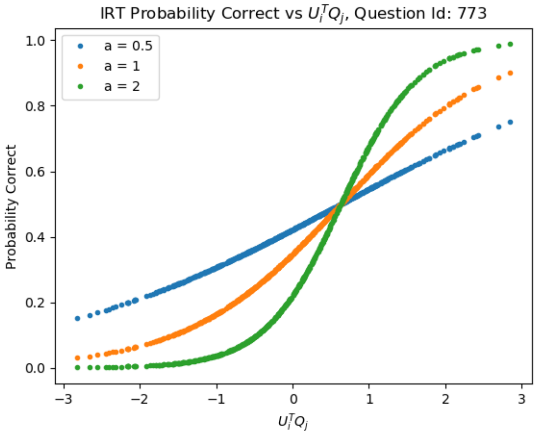
If we are trying to predict how this student will answer a linear algebra question, corresponding to a Q_j vector of:

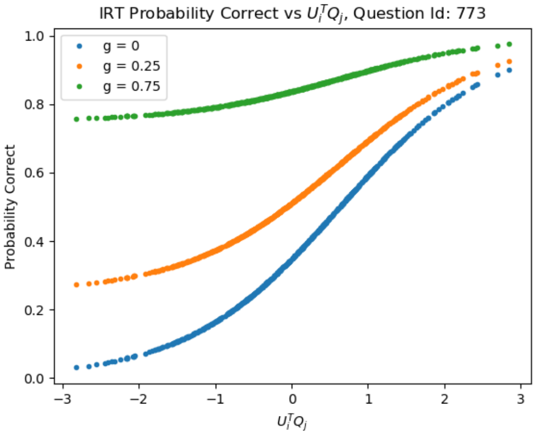
$$Q_j = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Then we can see that:

$$U_i^T Q_j = \begin{bmatrix} 1.5 & 1.5 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 1(1.5) + 0(1.5) + 1(0) = 1.5$$

As we can see, we do not incorporate the student's calculus ability into the probability they will get this question correct, since the question does not deal with calculus, but we are taking into account the student's general mathematical ability (represented by $U_{i,0}$).
We also felt that by adding the additional parameters a_j and g, we could capture a more nuanced relationship for each question's difficulty. Here, we show how changing a and g impacts the IRT distribution of question id 773.





We see that by changing the a value for question 773, the shape of the probability distribution changes. As a grows larger, the distribution takes on more of a steep sigmoidal shape, while smaller values of a represent a more flat sigmoidal curve, almost a linear relationship between a student's general ability on the question's categories and the probability that they answer the question correctly. On the other hand, by changing the value of g, we see that the basic shape of the distribution doesn't change all that much; instead, the range of the distribution is now compressed so that g is its floor.

In order to implement our algorithm, we will maximize the log likelihood and perform alternating gradient descent on the negative log likelihood.

Let $O = \{(n, m) : \text{entry } (n, m) \text{ of matrix } C \text{ is observed}\}$. Then we know that

$$p(\mathbf{C}|\mathbf{U}, \mathbf{Q}, \boldsymbol{\beta}, \mathbf{a}, g) = \prod_{i,j \in O} p(c_{i,j}|U_i, Q_j, \beta_j, a_j, g)^{c_{i,j}} (1 - p(c_{i,j}|U_i, Q_j, \beta_j, a_j, g))^{(1-c_{i,j})},$$

so

$$\log p(\mathbf{C}|\mathbf{U}, \mathbf{Q}, \boldsymbol{\beta}, \mathbf{a}, g) = \sum_{i,j \in O} c_{i,j} \log\left(\exp\left(a_j(U_i^T Q_j - \beta_j)\right) + g\right) - \log\left(\exp\left(a_j(U_i^T Q_j - \beta_j)\right) + 1\right) - (c_{i,j} - 1) \log(1 - g)$$

Then, we want to optimize $\mathbf{U}$, $\boldsymbol{\beta}$, and $\mathbf{a}$ using alternating gradient descent

$\frac{d}{dU_i} = \sum_{j:(i,j) \in O} a_j Q_j \exp(a_j U_i^T Q_j) \left(\frac{c_{i,j}}{g \exp(a_j \beta_j) + \exp(a_j U_i^T Q_j)} - \frac{1}{\exp(a_j \beta_j) + \exp(a_j U_i^T Q_j)} \right)$

$\frac{d}{d\beta_j} = \sum_{i:(i,j) \in O} a_j \exp(a_j U_i^T Q_j) \left(\frac{1}{\exp(a_j \beta_j) + \exp(a_j U_i^T Q_j)} - \frac{c_{i,j}}{g \exp(a_j \beta_j) + \exp(a_j U_i^T a_j)} \right)$

$\frac{d}{da_j} = \sum_{i:(i,j) \in O} \exp(a_j U_i^T Q_j)(U_i^T Q_j - \beta_j) \left(\frac{c_{i,j}}{g \exp(a_j \beta_j) + \exp(a_j U_i^T Q_j)} - \frac{1}{\exp(a_j \beta_j) + \exp(a_j U_i^T Q_j)} \right)$

Comparison

We posited that by introducing the two parameters a and g and the implementation of the category-based learning model, our model's bias would lower as it was better able to model the nuances present in the dataset. To test our hypotheses, we introduced each modification of our algorithm incrementally, testing the performance of each model at each step to measure the impact of every addition.

Classification Accuracies for Varying IRT Models

Model	Train	Validation	Test
baseline	0.7315657634772792	0.7090036692068868	0.7025119954840531
a	0.7337355348574655	0.7092859158904883	0.7036409822184589
a and g	0.7289197008185154	0.7098504092576913	0.6996895286480384
a, g, and category	0.7620836861416879	0.7053344623200677	0.6906576347727914

Adding the additional **a** parameter allowed us to model a more complex hypothesis space that fit the training set better than our base model. This led to an increase in performance in Train, Validation, and Test data sets.

However, adding the **g** parameter added an assumption to our model that did not seem to be reflected in the data set, as evidenced by the fact that we saw a higher validation accuracy after adding this parameter, but saw a decrease in accuracy on both the train and test data sets. Though we felt that this assumption of a minimum 25% chance of correctness on each question was valid given the multiple choice nature of the dataset, there could potentially be other factors involved that break down this assumption. These could include factors such as trick questions which skew the probabilities of selecting a correct answer, or the possibility of leaving a question unanswered (but still marked as incorrect).

Giving the model access to the question subject metadata allowed the model to fit a more complex relationship that fit the training set better. That said, we can see evidence that this model is overfitting on the training data, as we see a decrease in accuracy on both the validation and the test data despite the very high training accuracy.

Limitations

Since we are trying to capture a more complicated relationship between question subject and student ability, we are trying to learn 388 independent ability parameters for each student, which naturally means that each parameter has a smaller subset of the data to train on. This means that in order to accurately capture the relationships that we are aiming for, we will need a much larger amount of data. We believe that this is the greatest limitation of our model, as the limited data set that we are working with, as well as the inherent sparseness of the data, do not provide adequate information for us to learn such a large number of parameters accurately. In particular, the model is unable to learn the data set in a way that generalizes well, as is evidenced by the large discrepancy in its training and validation/test accuracies. With more data, the model would be more likely to learn broader trends, rather than focusing in on the training set's specific minutia.

Another limitation which is found in all of our existing models is that students have some inherent level of unpredictability in their answers. Even if a student has perfect knowledge of a subject, they may answer a question incorrectly. Our current model, and all the models that we have tested within this project, make the underlying assumption that the students are predictable, and that their answers to future questions can reliably be predicted using their previous performance and the performance of other students. To address this in our model, we could introduce another parameter s, which would be used to account for this inconsistency in human behaviour. With this parameter, our model would become:

$$p(c_{ij} = 1|\mathbf{U}_i, \beta_j, \mathbf{Q}_j, a_j, g) = g + \frac{(1 - g - s)\exp(a_j(\mathbf{U}_i^T\mathbf{Q}_j - \beta_j))}{1 + \exp(a_j(\mathbf{U}_i^T\mathbf{Q}_j - \beta_j))}$$

However, adding another parameter exacerbates our issue with the scarce data set, meaning it would likely not be worth it to add s.

Despite the hope that we would be able to better model the data set by adding more parameters, in the end, it turns out that we simply need more data for our methodology to succeed.

Works Referenced

Benedetto, Luca. "Item Response Theory for Assessing Students and Questions (Pt. 2)." Medium, January 18, 2020. https://medium.com/@lucabenedetto/item-response-theory-for-assessing-students-and-questions-pt-2-edf3d0c142b7.