542 students and 1774 different mathematical questions were sampled from a dataset given by Eedi, an online learning platform designed to be used in a classroom. Each student answered a subset of the questions and we were provided with data on whether they answered correctly, which we represent as a matrix of C. The challenge comes from the fact that because none of the students answered all of the questions, C ends up being a sparse matrix.

Model A

Let θ_i be a value that represents the ith student's ability and β_j be a value representing the difficulty of question j. We model the probability that student i answers question j correctly as $p(C_{ij} = 1) = exp(\theta_i - \beta_j)/(1 + exp(\theta_i - \beta_j))$. We then have that:

$$logp(C|\theta, \beta) = log(\prod_{i,j}(I(C_{ij} = 1)(exp(\theta_i - \beta_j))/(1 + exp(\theta_i - \beta_j))$$
$$+ I(C_{ij} = 0)(1 - exp(\theta_i - \beta_j)/(1 + exp(\theta_i - \beta_j)))))$$

$$= \sum_{i,j}(I(C_{ij} = 1)log(exp(\theta_i - \beta_j)/(1 + exp(\theta_i - \beta_j)))$$
$$+ I(C_{ij} = 0)log(1 - exp(\theta_i - \beta_j)/(1 + exp(\theta_i - \beta_j))))$$

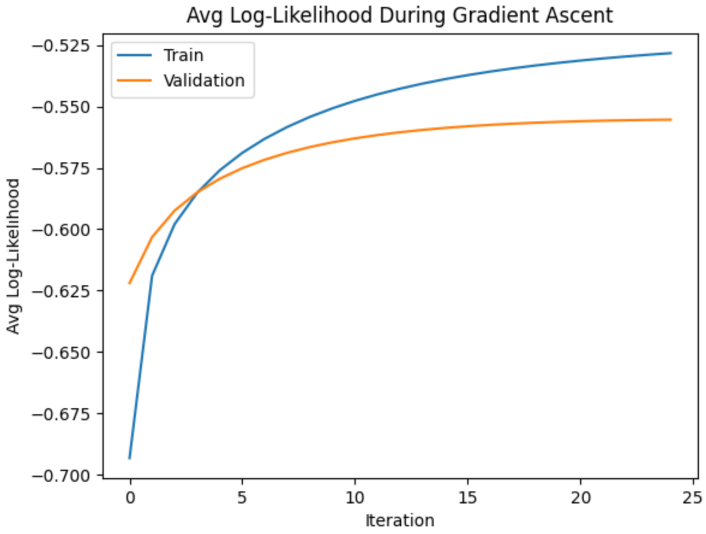
$$= \sum_{i,j}(I(C_{ij} = 1)((\theta_i - \beta_j) - log(1 + exp(\theta_i - \beta_j)))$$
$$+ I(C_{ij} = 0)log(1 - exp(\theta_i - \beta_j)/(1 + exp(\theta_i - \beta_j))))$$

Then, $\frac{\partial logp(C|\theta,\beta)}{\partial\theta_i} = \sum_j(I(C_{ij} = 1)(exp(\beta_j)/(exp(\theta_i) + exp(\beta_j)))$
$+ I(C_{ij} = 0)(- exp(\theta_i)/(exp(\theta_i) + exp(\beta_j))))$ and

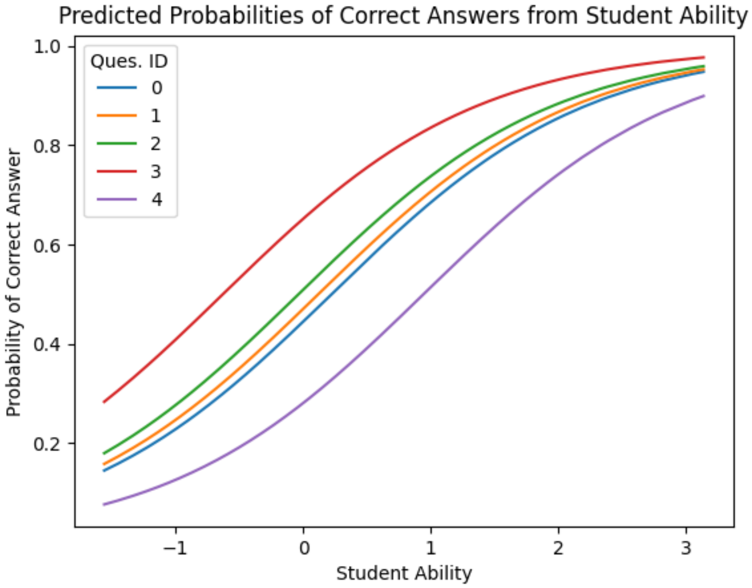
$$\frac{\partial logp(C|\theta,\beta)}{\partial\beta_j} = \sum_i(I(C_{ij} = 1)(- exp(\beta_j)/(exp(\beta_j) + exp(\theta_i)))$$
$$+ I(C_{ij} = 0)(exp(\theta_i)/(exp(\theta_i) + exp(\beta_j))))$$

For our model, we will simply denote θ_i as the proportion of questions student i answered correctly and β_j the proportion of students who answered question j correctly.

Training



Learning Rate: 0.01
Number of iterations: 25
Final Validation Accuracy: 0.7067
Final Test Accuracy: 0.7056



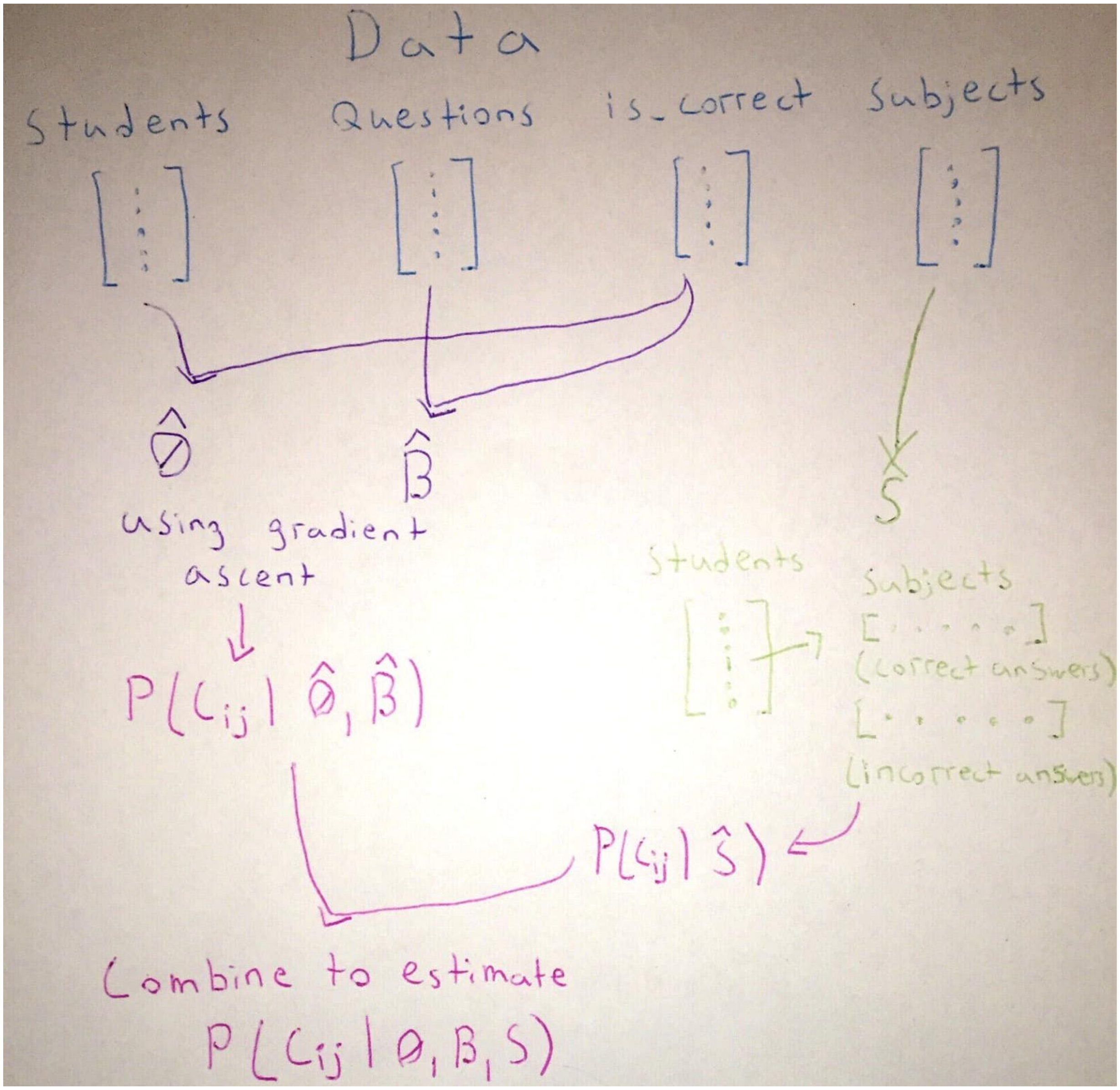
As expected, we see the probability of answering a given question correctly increases with measured student ability. Each of the plotted curves represents a sigmoid shape, which is how we modeled the probability of a correct answer.

Model B

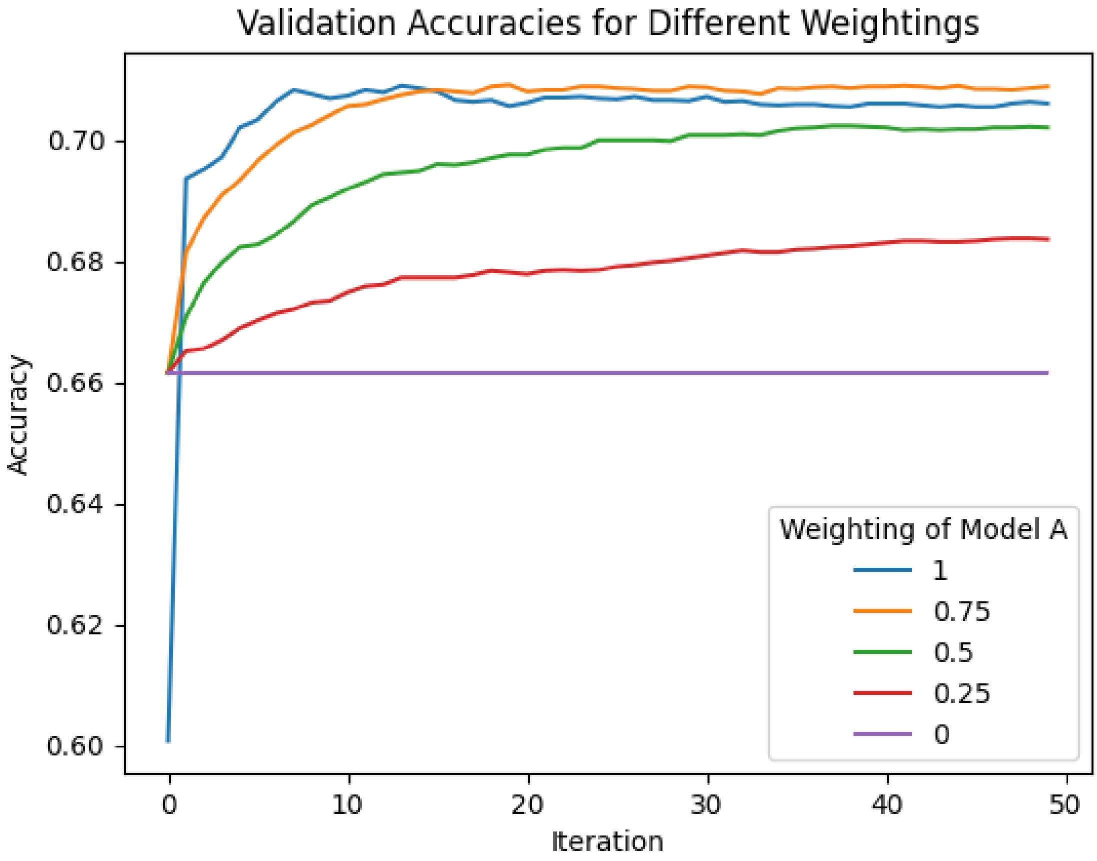
Model A may seem overly simplistic in that the proficiency of students usually varies across subject area. In fact, it is often the case that there are certain areas one is proficient in and others where one has trouble. For this reason, we assume the Model A is underfitting by not taking into account enough of the available data.

We now model $P(C_{ij} = 1 \mid \theta, \beta, S)$ for a particular student i and question j, where C is the sparse matrix and S is a parameter that denotes the proficiency of each student within each subject. We expect the bias to decrease in this latter model by taking into account the parameter S.

We do not need to convert to a generative model because we can already calculate $P(C_{ij} = 1 \mid \theta, \beta, S)$ without needing to explicitly find $P(S \mid \theta, \beta, C_{ij} = 1)$. So for the new model, we make a dictionary with question IDs as keys and tuples of the subject IDs as values. We then use this in conjunction with the training data to create a new dictionary where the keys are student IDs and the values are arrays that keep track of how many questions the corresponding student answered correctly and incorrectly for each subject. We train the $\theta\ and\ \beta$ parameters the same way we did in Part A, but now we have a new parameter S which we can use to aid in our predictions. Given student i and a new question j we are trying to determine C_{ij} for, we average their answer accuracy for each subject that question belongs to, which gives us a new prediction. We will try combining the prediction of our old Model A with the prediction given by Model S for different weights to see if we can arrive at a new Model B with improved accuracy.


Data
Students
Questions
is_correct
Subjects
$\hat{\theta}$
$\hat{B}$
$\hat{S}$
using gradient ascent
$P(C_{ij} | \hat{\theta}, \hat{B})$
Students
Subjects
(correct answers)
(incorrect answers)
$P(C_{ij} | \hat{S})$
Combine to estimate
$P(C_{ij} | \theta, B, S)$

Comparison/Demonstration



We see from the figure that Model A outperforms the mixed model for every set of weights tested, except when the weights on Model A and Model S are 0.75 and 0.25, respectively. This model, which we will adopt as Model B, has a higher accuracy than Model A past 16 iterations of gradient ascent. The final accuracy of Model B after 50 iterations is 0.70886 which is marginally greater than the accuracy of 0.70604 of Model A after 50 iterations.

By testing out different weights, we begin to see just what kind of effect Model S has on Model B as the weighting of the former increases. This way, we are able to assess if our hypothesis that Model B would decrease underfitting was correct.

We see that when the weighting on Model S is 1, the accuracy is constant with respect to the number of iterations. This is to be expected because in this case, the parameters $\theta\ and\ \beta$ are not used, so the gradient ascent has no effect on the model predictions. Our experiment tells us that we were wrong in our assumptions that Model B would have an increased accuracy and decrease overfitting because it seems as though the less we take into account Model S, the greater our accuracy until we reach our Model B, at which point the difference is negligible. Thus, we cannot conclude that Model B is a better model than Model A.

Limitations and Conclusion

Our modified model failed to improve upon our original one, as we couldn't find a set of weights that had more than a negligible effect on improving accuracy. Our results from the previous section suggest that a question's subject matter is relatively not relevant to predicting whether a student will answer it correctly. This was initially surprising, but we must consider the subject categories more closely. All the data came from Eedi, which only offers questions in mathematical areas. It seems fairly reasonable that mathematical ability can be well-generalized across mathematical sub-fields in that one would expect a student proficient with fractions to also be proficient at algebra, especially more so than a student proficient at fractions also be proficient with reading comprehension. Thus, it should be expected that proficiency in any particular subject would be highly correlated with a measure of a student's proficiency in general here, θ and the addition of parameter estimate $\hat{S}$ would not yield any useful information given that we already have an estimate for θ. Even if this were not the case, we notice that for many of the students, our $\hat{S}$ stored a fairly sparse array representing the proficiencies of the student. This means we have many subject categories, but perhaps not enough questions answered by students in each category to make the parameter estimate useful. This is why we combine the models rather than solely use Model S because we arrive at the problem of our Model S predictions being far too variable and sensitive to the effect of individual questions due to data sparsity and the large spread of possible subjects. Thus, one way we could address this problem is simply to collect more data. Additionally, if one wanted to predict correct answers to questions across the range of academic disciplines, perhaps our method would be more useful.