CSC311 Project

Pete Chen, Ocean Chen, Harvi Karatha

April 2024

1 Part A:

2 Question 1

2.1 Run User-based Collaborative Filtering

Test Accuracy for user-based CF with best k: 0.6841659610499576
Best k for user-based CF: 11

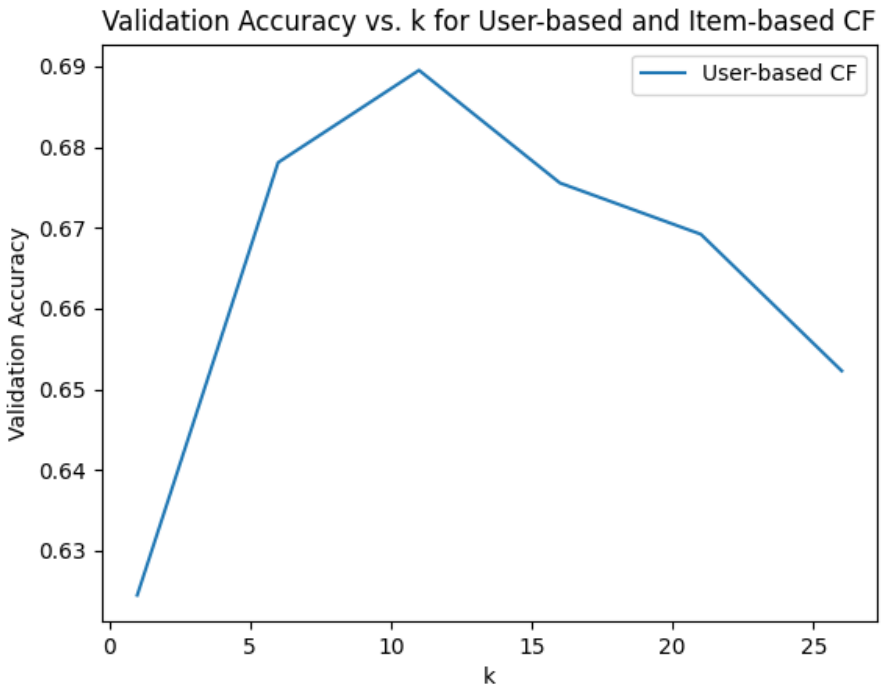


Figure 1: User-Based

2.2 Item-based Collaborative Filtering

The underlying assumption of item-based collaborative filtering is that if the patterns of correct and incorrect answers for Question B from students is similar to the pattern of answers for Question A then Question A is

```
Validation Accuracy: 0.6244707874682472
Validation Accuracy: 0.6780976573525261
Validation Accuracy: 0.6895286480383855
Validation Accuracy: 0.6755574372001129
Validation Accuracy: 0.6692068868190799
Best k (user-based): 11
Highest accuracy: 0.6895286480383855
```

Figure 2: User-Based Chosen K

similar to Question B. Therefore, the assumption is that Question A is similar to Question B in how students would respond to it.

2.3 Run Item-based Collaborative Filtering

Test Accuracy for item-based CF with best k: 0.6816257408975445
Best k for user-based CF: 21

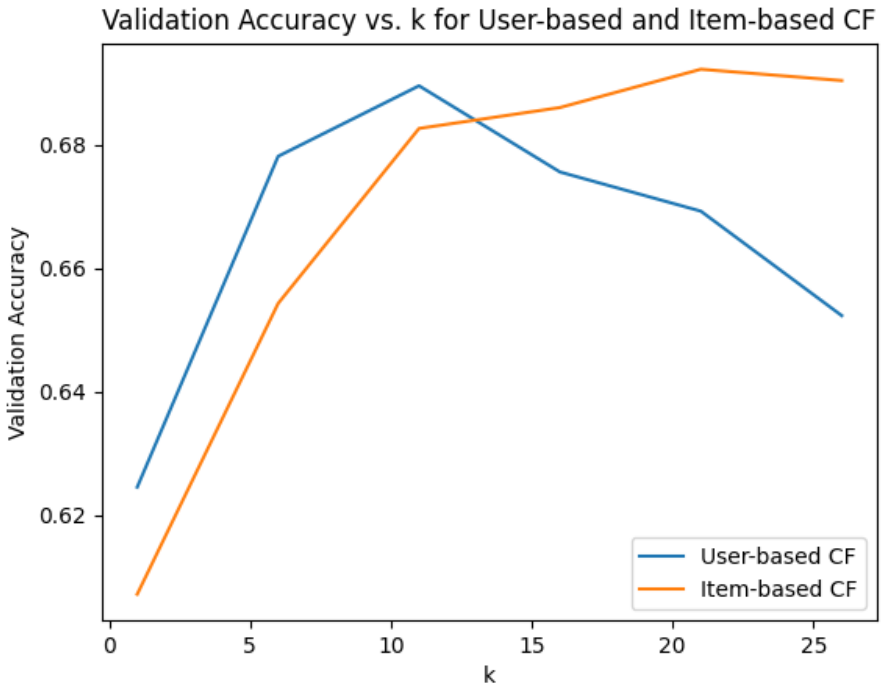


Figure 3: Item-Based

2.4 Comparison of User-Based & Item-Based Filtering

User-based collaborative filtering is the better method because the testing accuracy for the user-based filtering is around 0.003 better.

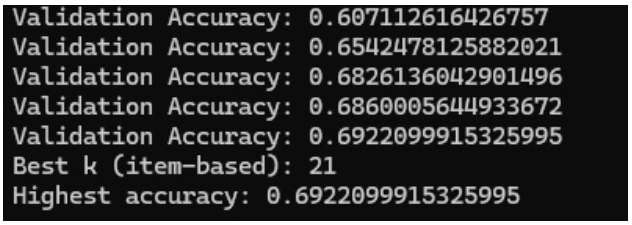


Figure 4: Item-based chosen k

2.5 Potential Limitations of kNN

There are 2 major limitations of using the kNN method for this need:
1. Time Complexity: due to the iteration method of kNN, it takes a lot more time to reach conclusions regarding whether a certain student will answer an arbitrary question correctly. KNN is very expensive in terms of time; its complexity is $O(nd)$, where n is the total number of data points and d is total number of features.
2. Fresh Data: With new questions or a new student, there might not be a sufficiently similar enough sample point from the data set to provide accurate answers with the kNN method.
3. Curse of Dimensionality: Since we are dealing with prediction based on the similarity of questions answered, the training data is actually in a very high dimension, and KNN may not be able to accurately output the result.

2.6 Question 2

2.6.1 Derivation of log-likelihood

Let n be the number of students, and m be the number of questions.

$$\log p(C|\theta,\beta) = \log \prod_{i}^{m}\prod_{j}^{n} p(c_{ij}|\theta_i,\beta_j) \tag{1}$$

$$= \sum_{i}^{n}\sum_{j}^{m} \log p(c_{ij}|\theta_i,\beta_j) \tag{2}$$

$$= \sum_{i}^{n}\sum_{j}^{m} c_{ij}\log(\frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}) + (1-c_{ij})\log(\frac{1}{1+\exp(\theta_i-\beta_j)}) \tag{3}$$

$$= \sum_{i}^{n}\sum_{j}^{m} c_{ij}(\theta_i-\beta_j-\log(1+\exp(\theta_i-\beta_j))) + (1-c_{ij})(-\log(1+\exp(\theta_i-\beta_j))) \tag{4}$$

$$= \sum_{i}^{n}\sum_{j}^{m} c_{ij}(\theta_i-\beta_j) - \log(1+\exp(\theta_i-\beta_j)) = L \tag{5}$$

Beside:

$$p(c_{ij}|\theta_i,\beta_j) \tag{6}$$

$$= p(c_{ij}=1|\theta_i,\beta_j)^{c_{ij}} * p(c_{ij}=0|\theta_i,\beta_j)^{1-c_{ij}} \tag{7}$$

$$= \frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}^{c_{ij}} * \frac{1}{1+\exp(\theta_i-\beta_j)}^{1-c_{ij}} \tag{8}$$

Derivatives with respect to θ_i:

$$\frac{\partial L}{\partial \theta_i} = \frac{\partial}{\partial \theta_i}\sum_{i}^{n}\sum_{j}^{m} c_{ij}(\theta_i-\beta_j) - \log(1+\exp(\theta_i-\beta_j)) \tag{9}$$

$$= \sum_{j}^{m}\frac{\partial}{\partial \theta_i}\sum_{i}^{n} c_{ij}(\theta_i-\beta_j) - \log(1+\exp(\theta_i-\beta_j)) \tag{10}$$

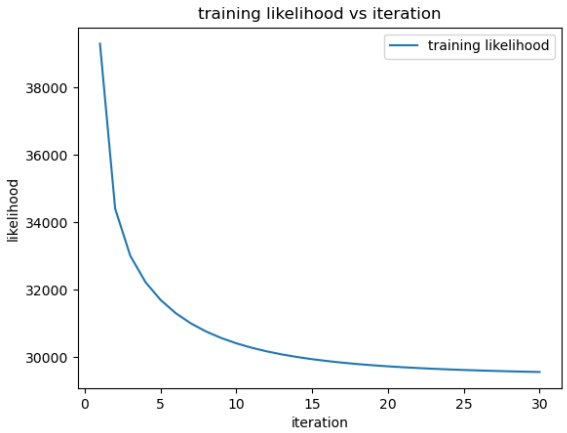
$$= \sum_{j}^{m}\left(c_{ij} - \frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)}\right) \tag{11}$$

Derivatives with respect to β_j:

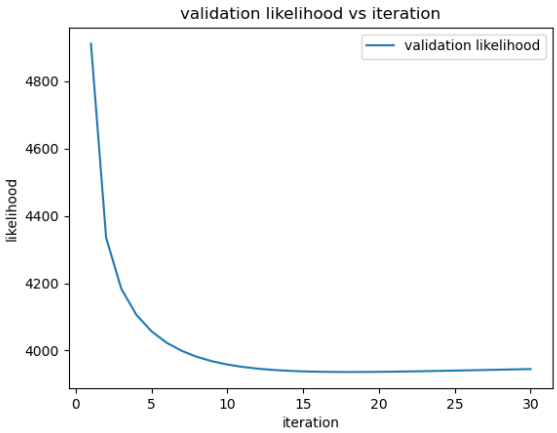
$$\frac{\partial L}{\partial \beta_j} = \frac{\partial}{\partial \beta_j}\sum_{i}^{n}\sum_{j}^{m} c_{ij}(\theta_i-\beta_j) - \log(1+\exp(\theta_i-\beta_j)) \tag{12}$$

$$= \sum_{i}^{n}\frac{\partial}{\partial \beta_j}\sum_{j}^{m} c_{ij}(\theta_i-\beta_j) - \log(1+\exp(\theta_i-\beta_j)) \tag{13}$$

$$= \sum_{i}^{n} -c_{ij} + \frac{\exp(\theta_i-\beta_j)}{1+\exp(\theta_i-\beta_j)} \tag{14}$$



(a) Training LLD



(b) Validation LLD

Figure 5: Iteration vs. Accuracy

2.6.2 Hyperparameter Tuning

Based on continuous testing, the chosen hyperparameters are:

Learning rate = 0.018

Iterations = 30

Test accuracy: 0.7075924357888794

Validation accuracy: 0.7070279424216765

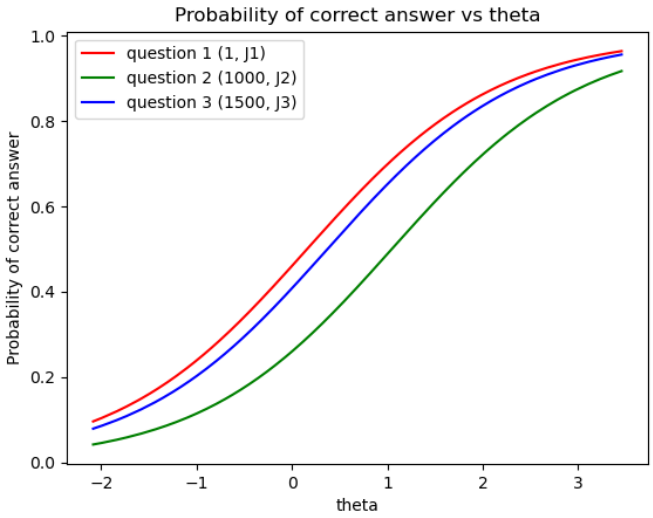


Figure 6: J1, J2, J3

2.6.3 Analyzing 3 Questions

$j1$ is the first question, $j2$ is the 1000th question, and $j3$ is the 1500th question.

$j1$ has difficulty level (β) of 0.15888388227768213.
$j2$ has difficulty level (β) of 1.0465154218689983.
$j3$ has difficulty level (β) of 0.37210046696004195.

In Figure 6, the red curve corresponding to question $j1$ is positioned above the others, suggesting it is the question most likely to be answered correctly by students. It illustrates that students have a greater chance of successfully responding to $j1$ relative to the two tougher questions, aligning with the β values where $j1$ has the smallest beta value. It is also true for $j2$ and $j3$.

The overall shape of the three curves corresponds to the shape of a sigmoid function because that we used a sigmoid activation function with the input being $\theta_i - \beta_j$ in the IRT model: $\frac{\exp(\theta_i - \beta_j)}{1+\exp(\theta_i - \beta_j)}$.

2.7 Question 3:

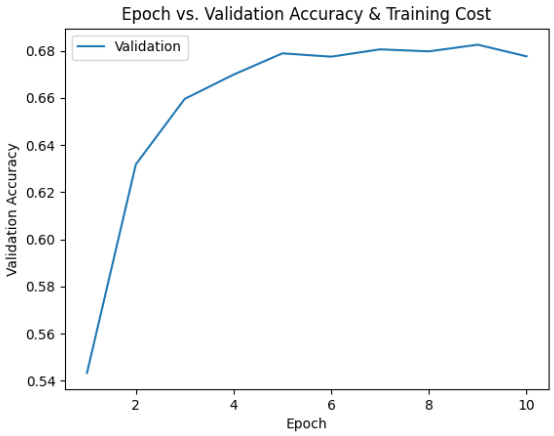
2.7.1 Differences between ALS and Neural Networks

There are three differences between the alternating least squares method and the neural networks method:

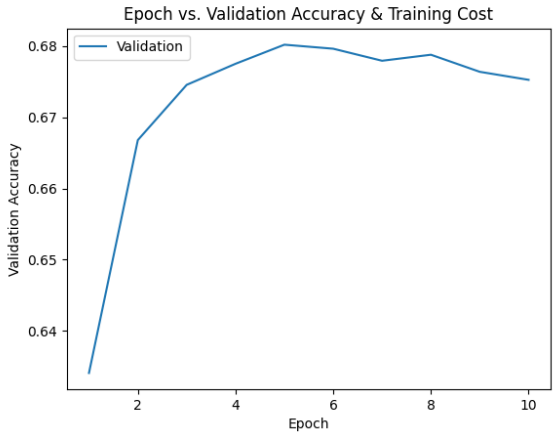
1. Complexity - Neural networks use forward and backward propagation to learn more complex processes and consist of interconnected layer that create a more complex model. With ALS, it iterates to minimize restructuring and creates a less complex model.
2. Training Process - Neural networks by readjusting weights back and forth in the model to minimize a loss function. Additionally, ALS uses gradient methods to minimize restructuring through iterations.
3. Large Datasets - Neural networks have an easier time handling larger datasets. ALS requires way larger matrices that can pose to become a more time consuming process compared to neural networks.

2.7.2 Choosing a K & Hyper Parameters

After choosing arbitrary parameters (learning rate = 0.01 and number of epochs = 10), we tried varied k values to see what works best.

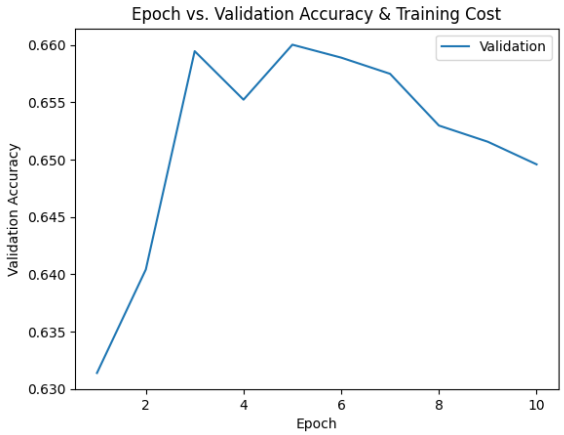


(a) K = 5

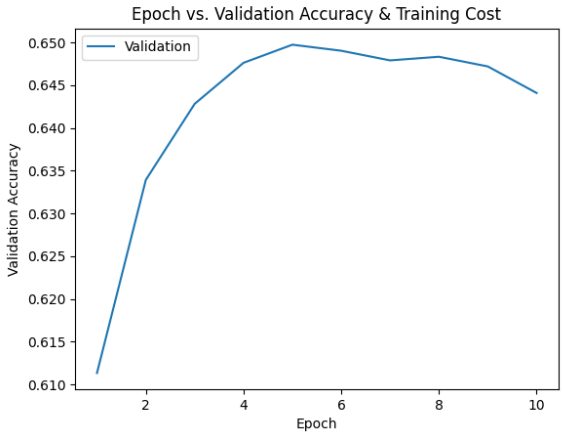


(b) K = 10

Figure 7: K Comparisons

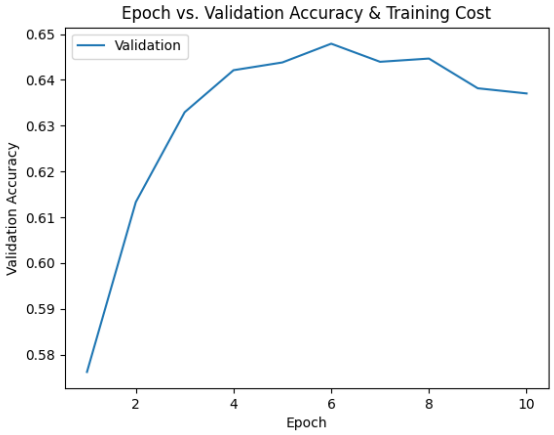


(a) K = 50

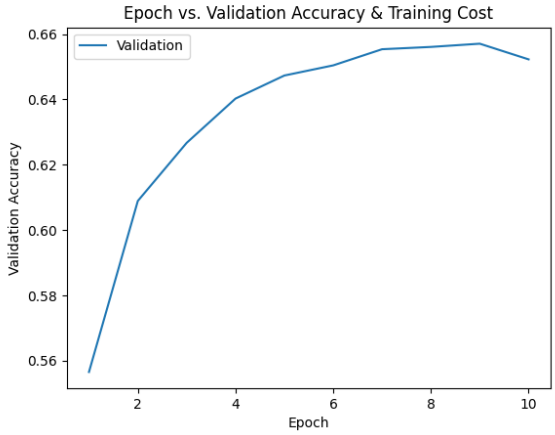


(b) K = 100

Figure 8: K Comparisons



(a) K = 200



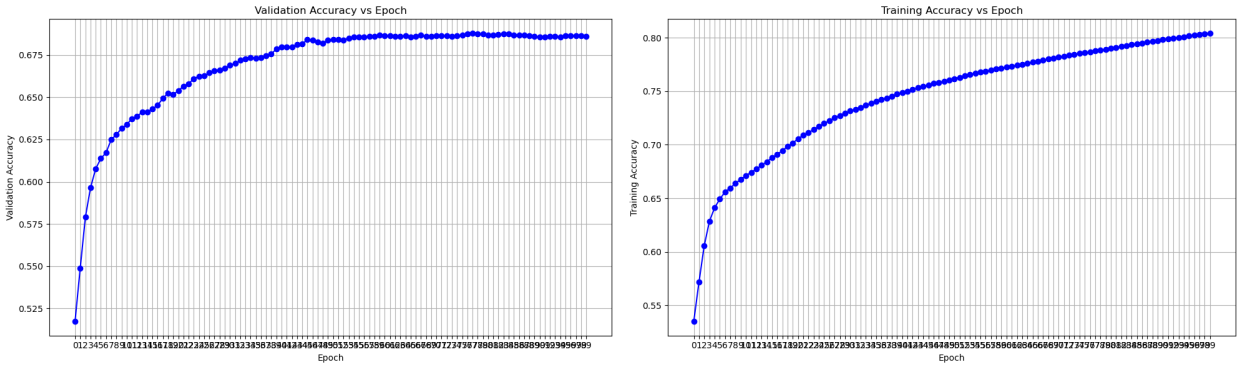
(b) K = 500

Figure 9: K Comparisons

After testing with varied learning rates and number of epochs, the following was the optimal combination:
Best K: 10
Best learning rate: 0.01
Best number of epoch: 100
Test Accuracy: 0.6923511148744003
Validation Accuracy: 0.6860005644933672

2.7.3 Training & Validation Objectives

Given that test accuracy comes out to the following: 0.6923511148744003



(a) Validation Accuracy VS Epoch

(b) Training Accuracy VS Epoch

Figure 10: Epoch vs. Accuracy

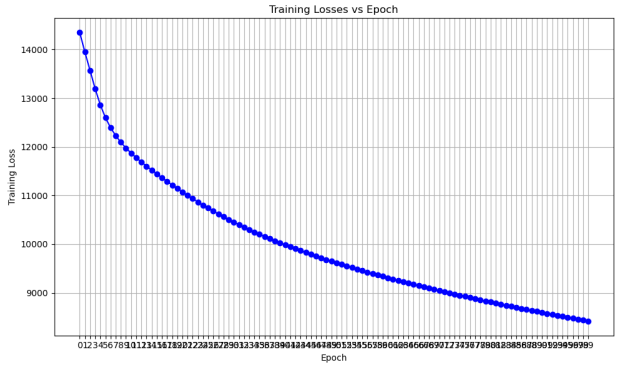


Figure 11: Training Loss VS Epoch

2.7.4 L_2 Regularization

Lambda	Validation Accuracy
0.001	0.687694
0.01	0.666667
0.1	0.653401
1	0.623201

Table 1: L_2 Regularization w/ Varied Lambdas

With chosen lambda = 0.001, the results are:
Test Accuracy: 0.6833192209991532
Validation Accuracy: 0.687694044594976
The validation accuracy is slightly higher than the accuracy from the model without regularizer.

3 Part B:

In Section B, We are exploring possible enhancements to the one-parameter IRT model presented in Section A. From reading Nathan Thompson's blog [1], we learned about the three-parameter IRT model, which offers advancements over the existing model by adding extra parameters.

3.1 Key Intuition

Currently, the one-parameter IRT incorporates with β which representing the difficulty of each question, we will add a discrimination parameter to account for how well an item differentiates between students of different abilities, and a pseudo-guessing parameter to represent the probability of guessing the question correctly purely by chance by the fact that there's always a chance that a student might guess the answer correctly in multiple-choice questions [2].

3.2 Why Would It Have Better Performance?

The Discrimination Parameter (α): The discrimination parameter allows each question to have a different sensitivity to the abilities of the student. This means that the model can distinguish between questions that are good at differentiating between high and low ability students and those that are not, which further allows it to better fit the data (avoid under-fitting).

The Guessing Parameter (γ): There's always a chance that a student might guess the answer correctly in multiple-choice questions; by adding a guessing parameter, the model would better fit the data by considering the case of guessing the answer. This can add more credibility to the model by considering some data points (which might better fit the data than the one-parameter IRT model without a guessing parameter).

3.3 Definition

θ_i: Student i's ability
β_j: The difficulty level of question j
α_j: The discrimination parameter for question j (how well a question differentiates between students of different abilities)
γ_j: Probability of correctly answering question j by random picking

3.4 Probability Formula ($c_{ij} = 1$):

$$p(c_{ij} = 1|\theta_i, \beta_j, \gamma_j, \alpha_j) = \gamma_j + (1 - \gamma_j)\frac{\exp(\alpha_j(\theta_i - \beta_j))}{1 + \exp(\alpha_j(\theta_i - \beta_j))}$$

This formula can be reduced to One-parameter IRT probability formula:
When α is the same for all questions (there is no difference between each pair of question and student's abilities) and guessing probability is 0 ($\gamma = 0$), the probability formula is exactly the same as our One-parameter IRT probability formula.

3.5 Formula V.2: See Calculation in Appendix

3.5.1 Log-Likelihood

$$
\begin{aligned}
\log p(C|\theta,\beta,\alpha,\gamma) &= \log \prod_{i=1}^{n}\prod_{j=1}^{m} p(c_{ij}|\theta_i,\beta_j,\alpha_j,\gamma_j) \\
&= \log \prod_{i=1}^{n}\prod_{j=1}^{m} (\gamma_j + (1-\gamma_j)\cdot \frac{\exp(\alpha_j(\theta_i-\beta_j))}{1+\exp(\alpha_j(\theta_i-\beta_j))})^{c_{ij}} \\
&\qquad \cdot (1-\gamma_j-(1-\gamma_j)\cdot \frac{\exp(\alpha_j(\theta_i-\beta_j))}{1+\exp(\alpha_j(\theta_i-\beta_j))})^{1-c_{ij}} \\
&= \log \prod_{i=1}^{n}\prod_{j=1}^{m} (\frac{\gamma_j+\exp(\alpha_j(\theta_i-\beta_j))}{1+\exp(\alpha_j(\theta_i-\beta_j))})^{c_{ij}} \cdot (\frac{1-\gamma_j}{1+\exp(\alpha_j(\theta_i-\beta_j))})^{c_{ij}} \\
&= \log \prod_{i=1}^{n}\prod_{j=1}^{m} \frac{(\gamma_j+\exp(\alpha_j(\theta_i-\beta_j)))^{c_{ij}}\cdot(1-\gamma_j)^{1-c_{ij}}}{(1+\exp(\alpha_j(\theta_i-\beta_j)))^{c_{ij}+(1-c_{ij})=1}} \\
&= \sum_{i=1}\sum_{j=1} \log(numerator) - \log(Denominator) \\
&= \sum_{i=1}\sum_{j=1} c_{ij}\log(\gamma_j+\exp(\alpha_j(\theta_i-\beta_j))) + (1-c_{ij})\log(1-\gamma_j) - \log(1+\exp(\alpha_j(\theta_i-\beta_j))) = L
\end{aligned}
$$

3.5.2 Derivative with respect to θ_i

$$
\begin{aligned}
&\frac{\partial L}{\partial \theta_i} \\
&\qquad = \sum_{j=1}^{m} c_{ij}\cdot\alpha_j\cdot\frac{\exp(\alpha_j(\theta_i-\beta_j))}{\gamma_j+\exp(\alpha_j(\theta_i-\beta_j))} - \alpha_j\frac{exp(\alpha_j(\theta_i-\beta_j))}{1+exp(\alpha_j(\theta_i-\beta_j))}
\end{aligned}
$$

3.5.3 Derivative with respect to β_j

Negation of $\frac{\partial L}{\partial \theta_i}$:

$$
\begin{aligned}
\frac{\partial L}{\partial \beta_j} &= -\frac{\partial L}{\partial \theta_i} \\
&= \sum_{i=1}^{m} -c_{ij}\cdot\alpha_j\cdot\frac{\exp(\alpha_j(\theta_i-\beta_j))}{\gamma_j+\exp(\alpha_j(\theta_i-\beta_j)} + \alpha_j\frac{exp(\alpha_j(\theta_i-\beta_j))}{1+exp(\alpha_j(\theta_i-\beta_j))}
\end{aligned}
$$

3.5.4 Derivative with respect to α_j

Similar to $\frac{\partial L}{\partial \theta_i}$ but coefficient α_j becomes $(\theta_i-\beta_j)$

$$
\begin{aligned}
&\frac{\partial L}{\partial \alpha_j} \\
&\qquad = \sum_{i=1}^{m} c_{ij}\cdot(\theta_i-\beta_j)\cdot\frac{\exp(\alpha_j(\theta_i-\beta_j))}{\gamma_j+\exp(\alpha_j(\theta_i-\beta_j)} - (\theta_i-\beta_j)\cdot\frac{exp(\alpha_j(\theta_i-\beta_j))}{1+exp(\alpha_j(\theta_i-\beta_j))}
\end{aligned}
$$

3.5.5 Derivative with respect to γ_j

$$\frac{\partial L}{\partial \gamma_j} = \sum_{i=1} c_{ij} \frac{1}{\gamma_j + \exp(\alpha_j(\theta_i - \beta_j))} + \frac{c_{ij} - 1}{1 - \gamma_j}$$

Beside:

$$\frac{\partial}{\partial \gamma_j}(\gamma_j + \exp(\alpha_j(\theta_i - \beta_j))) = 1 + 0 = 1$$

$$\frac{\partial}{\partial \gamma_j} \log(1 + \exp(\alpha_j(\theta_i - \beta_j))) = 0$$

3.6 Implementation:

We've implemented both 3PL (α, θ, γ, β) and 2PL (α, θ, β).
See 3PL.py and 2PL.py

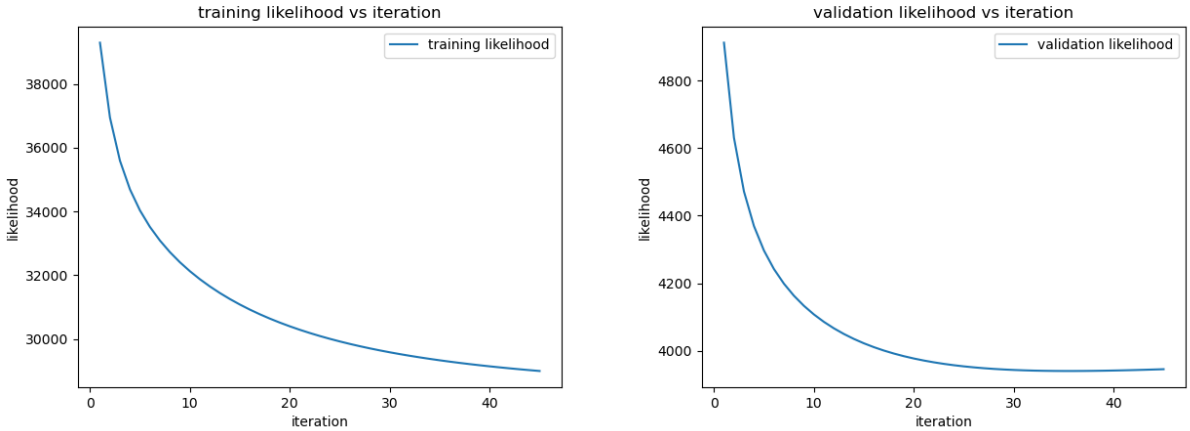
3.7 2PL:

Learning Rate	Iterations
0.005	45

Table 2: 2PL Hyperparameter

Test Accuracy	Validation Accuracy
0.7056167090036692	0.7082980524978831

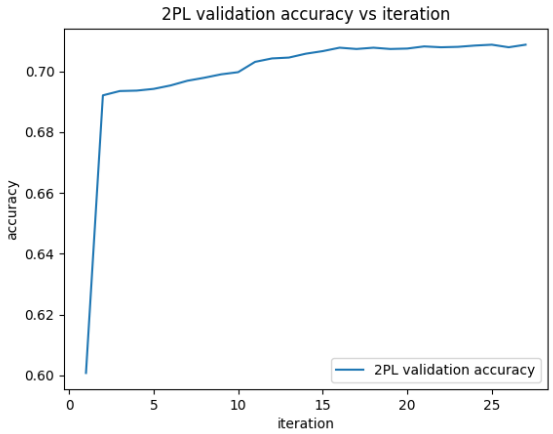
Table 3: 2PL Accuracy



(a) Training LLD vs. Iterations

(b) Validation LLD vs. Iterations

Figure 12: 2PL



(a) Validation Accuracy vs. Iterations

Figure 13: 2PL validation accuracy

3.8 3PL:

Learning Rate 2(γ)	Learning Rate	Iterations
0.0001	0.005	45

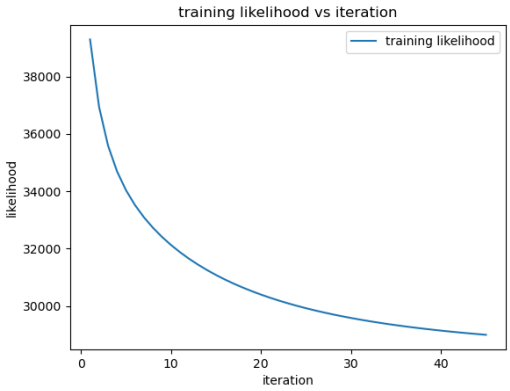
Table 4: 3PL Hyperparameter

Test Accuracy	Validation Accuracy
0.7036409822184589	0.707874682472481

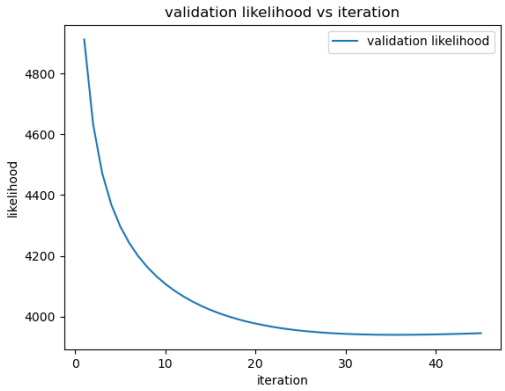
Table 5: 3PL Accuracy

Average γ	Average α
-0.003195812413666688	1.2055388800027158

Table 6

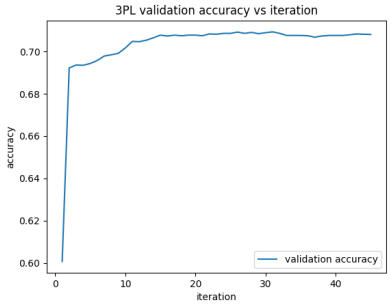


(a) Training LLD vs. Iterations



(b) Validation LLD vs. Iterations

Figure 14: 3PL



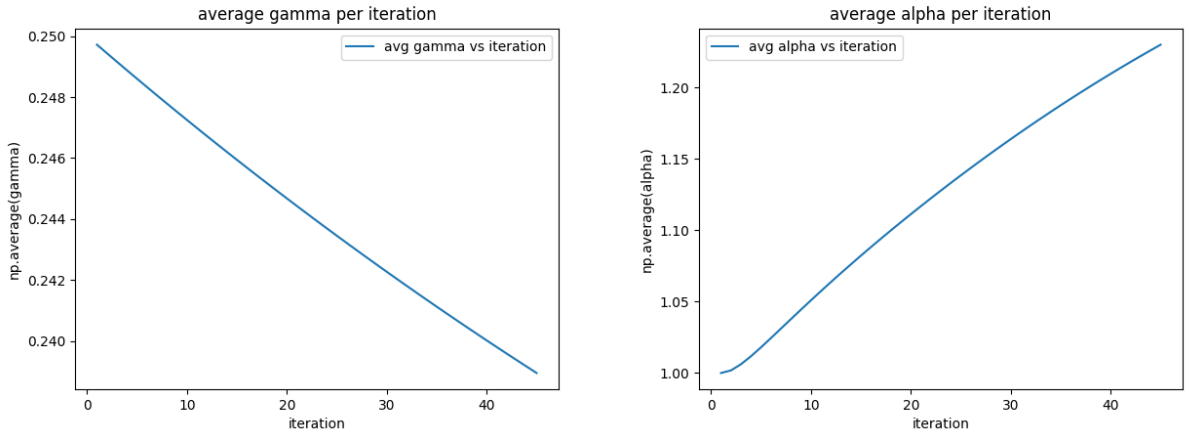
(a) Validation Accuracy vs. Iterations

Figure 15: 3PL validation accuracy

3.9 Problem with γ (the guessing parameter):

While tuning the hyper-parameters for the 3PL and 2PL models, we've noticed that omitting γ (2PL) performs consistently better in terms of validation accuracy even when trained under the same conditions. This suggests that the guessing, and by extension γ, has a negligible impact on the accuracy of students' answers in our dataset.

To further this hypothesis, we've compared the average values of the discrimination parameter α against that of γ, and found that the final average value of γ (-0.003196) is less than 0.3% of α (1.20055). This could possibly be because γ is initialized to a 0 array, but even when initialized to a non-zero values, γ still exhibits a pattern of convergence towards 0, see figures 14.



(a) γ initialized to 0.25 vs. Iterations

(b) α vs. Iterations

Figure 16: Gamma vs. Alpha per iteration

The above analysis suggests that parameter γ likely does not add any significant explanatory power to the IRT model, while also supporting the importance of the discrimination parameter α.

The reason behind the lack of significance in γ may be due to several factors.

Characteristics of Diagnostic Questions:
Given that our dataset is sourced from diagnostic questions specifically designed to highlight common misconceptions, the role of guessing may be minimized. This could be because diagnostics on Eedi may be set up in a way such that incorrect answers are purposefully designed to capture specific misconceptions rather than being randomly chosen. In such a case, even if students were to guess, the chances of a guess leading to a correct answer are minimal, thus diminishing the utility of modeling guessing behavior. If most students are choosing answers based on misconception rather than guessing, the relevance of γ is significantly reduced, making 2PL much more effective in our context.

Overfitting:
The addition of γ increases the complexity of our model, and might lead to overfitting. As a result, the model might have trouble generalizing to the validation dataset, reducing its predictive performance.

3.10 Your Model versus Baseline Models

Model	Validation Acc.	Test Acc.
KNN	0.69220999	0.6816257408975445
NN	0.687694044594976	0.6833192209991532
IRT	0.7070279424216765	0.7075924357888794
2PL	0.7082980524978831	0.7056167090036692
3PL	0.707874682472481	0.7036409822184589

Table 7

As shown above, 2PL generates the highest validation accuracy among all models.

.

3.11 Overall Model's Performance

Both 3PL and 2PL (the extended IRTs) consistently performed better than the baseline one-parameter IRT model, with 2PL seeing the greatest performance increase. Although 3PL incurred a slight increase in validation accuracy over 1PL, the performance boost is strictly attributed towards the discrimination parameter α. Empirical evidence showed that the introduction of the additional guessing parameter γ was counter-intuitive, and 2PL was not only sufficient, but consistently superior (*see analysis in section 3.9*).

While section 3.9 exposes γ's irrelevance, it also highlighted the importance of α in comparison. The correlation between the magnitude of alpha and validation accuracy suggests that the ability of questions to discriminate between students of varying abilities is a significant factor in explaining student performance. This could be due to the nature of the questions in our dataset, which are purposely designed to identify specific misconceptions, so the ability to differentiate between students is crucial. Since the majority of questions may be designed to pinpoint exactly where students' understanding begins to diverge based on their responses, the addition of α can aid in capturing these nuances in our model.

In terms of a possible method of improving the model's performance, using more datasets to better tune the hyper-parameters would aid in creating a more widely applicable model. Another alternative method of having a more widely usable model would be to have a hyperparameter choosing process built into the code that chooses the best hyperparameters based on the relevant, provided dataset. This would better account for skews and biases in provided datasets and better fit the model to those differences.

3.12 Possible Limitations of Model

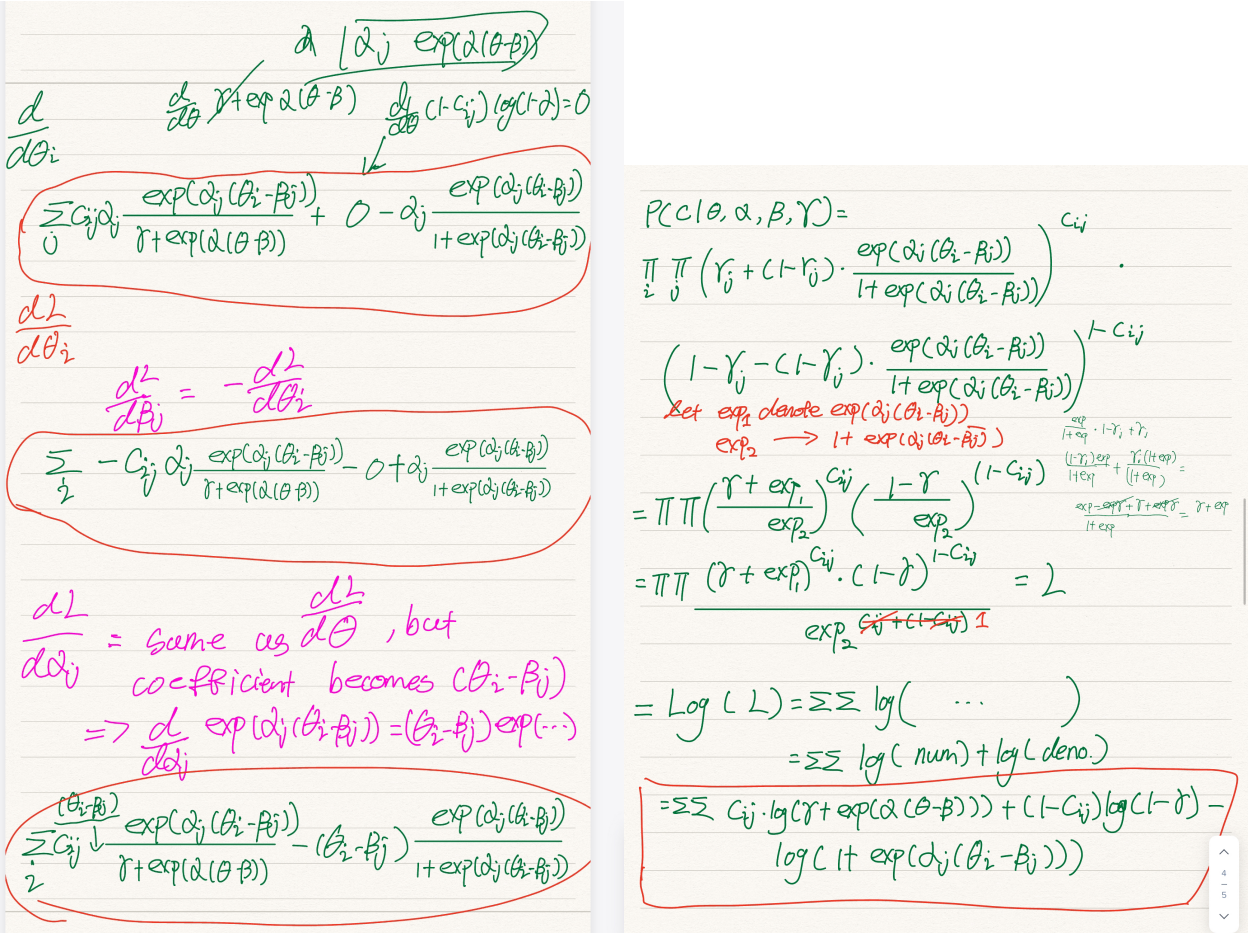
One limitation of the model is the possibility of covariances between the chosen factors. For example, there is likely a correlation between the the discrimination factor and the difficulty factor which might have an impact on the model as it could give more accurate decisions upon considering how one impacts another. More explicitly, how well a question differentiated students of varied abilities is likely informed and impacted by the difficulty of the question itself. One setting where this can come up is a very easy question that differentiated almost no students, which can show an important weighting of both factors without realizing that the reason behind the lack of differentiation is because of the low difficulty of the question. A possible modification to address this limitation would be to include another variable to model the correlation factor between α and β.

4 Member Contributions

Jiahao, Ocean, and Harvi make up the team members for this project. As a team, each individual of the team completed the three questions in Part A on their own (and all chose to implement the Neural Network process for Question 3). Since everyone implemented Part A seperately, Ocean and Jiahao worked

on writing out the derivatives in the report for the Part B model where Harvi wrote out the information about the possible limitations and overall model performance in the report. All three members implemented Part B model through code using their own processes and cross-compared answers when trying out new hyperparameters for the Part B model. Where Jiahao and Ocean found that the 2 factor model was more effective and continued testing varied parameters and models that were added as images in the report, Harvi worked on writing out the Part A - Question 1 & 3 content in the report. In the end, the most efficient and concise code for each question in Part A out of each of the three options created by each of the members was submitted.

5 Appendix:



(a) (b)

Figure 17

References

[1] Thompson, Nathan, PhD. "What Is the Three Parameter IRT Model (3PL)?" Assessment Systems, 28 Feb. 2024, assess.com/three-parameter-irt-3pl-model.

[2] "Item Response Theory." Wikipedia, 18 Feb. 2024, en.wikipedia.org/wiki/Item_response_theory.