CSC311 Final Project

Ian Huang, Benjamin Liu

Part A

1. ***k*-Nearest Neighbor**

   a. See Figure 1 and Table 1.

   b. See Table 1.

   c. See Table 1. The underlying assumption for item-based similarity is that for two questions **a** and **b** with answers $a_1, \ldots, a_n$ and $b_1, \ldots, a_n$ for each student $i \in \{1, \ldots, n\}$, if $a_i = b_i$ for most $i \neq j$, then it is likely that $a_j = b_j$ as well.

   d. k-NN by user distance marginally outperformed k-NN by item distance based on the resulting test accuracies for the respective experimentally derived values for k^*.

   e. There are a number of potential limitations with this method. For instance, the assumption described above for item-based filtering and the similar assumption for user-based filtering may not hold. Additionally, another potential problem with this method is that of data scarcity. The sparse matrix from which item/user distance is computed is around 94% missing values[1], which may limit the efficacy of the underlying NaN-Euclidean distance metric used in the k-NN implementation.

k	`knn_impute_by_user`	`knn_impute_by_item`
1	0.624	0.607
6	0.678	0.654
11	**0.690**	0.683
16	0.676	0.686
21	0.669	**0.692**

Table 1: Data represented in Figure 1. Validation accuracy with respect to k for distance by user and by item. The highest validation accuracies were obtained by $k^* = 11$ and $k^* = 21$ for user and input similarity respectively for which the test accuracies were 0.684 and 0.682 respectively.

2. **Item Response Theory**

   a. Given that

$$z \overset{\text{def}}{=} p(c_{ij} = 1 \mid \theta_i, \beta_j) = \frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)},$$

[1] Calculated by `np.count_nonzero(np.isnan(sparse_matrix)) / sparse_matrix.size`

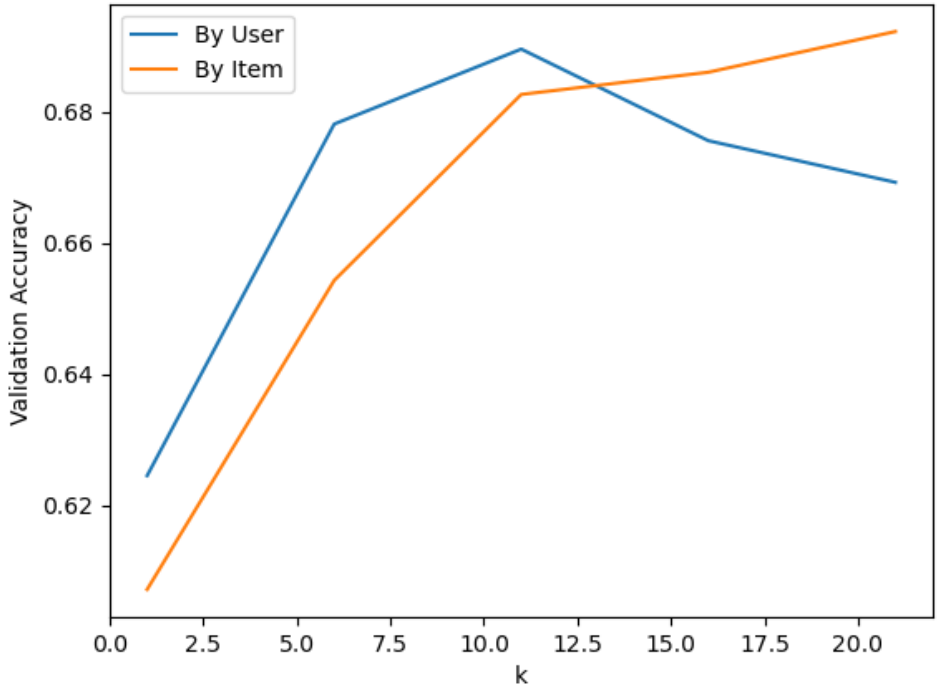


Figure 1: Validation accuracy with respect to k.

the (positive) log-likelihood $\log p(c_{ij} \mid \theta_i, \beta_j)$ is given by:

$$\begin{aligned} \ell &\stackrel{\text{def}}{=} \log p(c_{ij} \mid \theta_i, \beta_j) \\ &= \log\left(z^{c_{ij}}(1-z)^{1-c_{ij}}\right) \\ &= c_{ij}\log(z) + (1-c_{ij})\log(1-z). \end{aligned} \tag{1}$$

Using (1), we have

$$\begin{aligned} \log p(\mathbf{C} \mid \boldsymbol{\theta}, \boldsymbol{\beta}) &= \log \prod_i \prod_j p(c_{ij} \mid \theta_i, \beta_j) \\ &= \sum_i \sum_j \log p(c_{ij} \mid \theta_i, \beta_j) \\ &= \sum_i \sum_j \ell = \sum_i \sum_j c_{ij}\log(z) + (1-c_{ij})\log(1-z). \end{aligned}$$

The partial derivatives of ℓ with respect to θ_i and β_j are given by[2]

$$\begin{aligned}\frac{\partial \ell}{\partial \theta_i} &= \frac{\partial \ell}{\partial z}\frac{\partial z}{\partial \theta_i} \\ &= \left(\frac{c_{ij}}{z} - \frac{1-c_{ij}}{1-z}\right) z(1-z) \\ &= c_{ij}(1-z) - z(1-c_{ij}) \\ &= c_{ij} - c_{ij}z - z + c_{ij}z \\ &= c_{ij} - z,\end{aligned} \qquad \begin{aligned}\frac{\partial \ell}{\partial \beta_j} &= \frac{\partial \ell}{\partial z}\frac{\partial z}{\partial \beta_j} \\ &= \left(\frac{c_{ij}}{z} - \frac{1-c_{ij}}{1-z}\right)(-z(1-z)) \\ &= \dots \\ &= -c_{ij} + z.\end{aligned}$$

b. The parameters $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ were randomly initialized element-wise using a uniform distribution over $[-0.1, 0.1]$ and optimized using the update rule[3]

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \frac{\partial \ell}{\partial \boldsymbol{\theta}} \quad \boldsymbol{\beta} \leftarrow \boldsymbol{\beta} + \alpha \frac{\partial \ell}{\partial \boldsymbol{\beta}}$$

with learning rate $\alpha = 0.001$ over 50 iterations. See Figures 2 and 3 for detailed results.

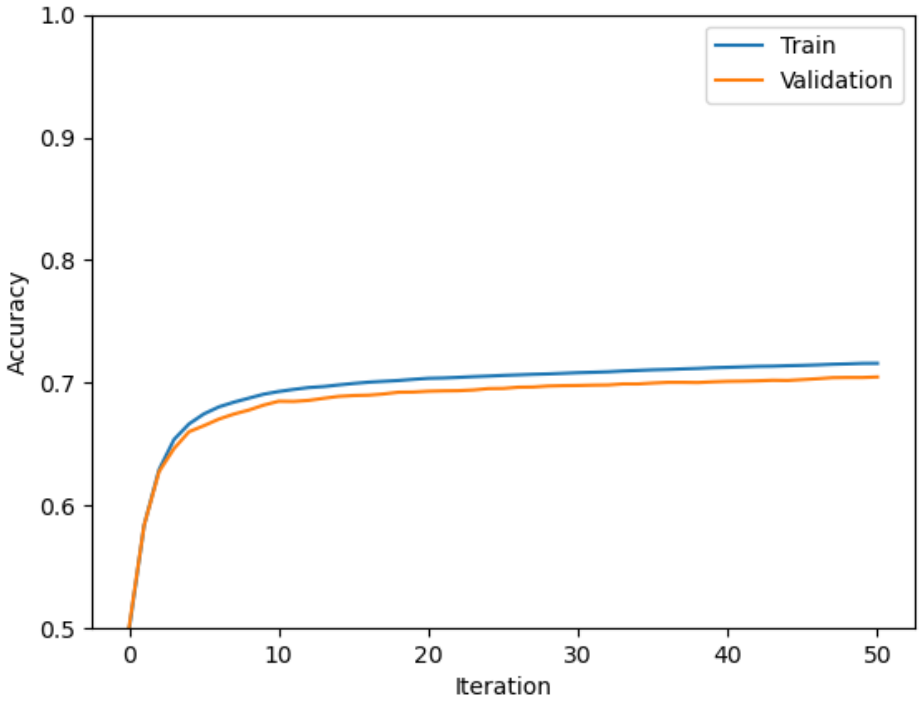
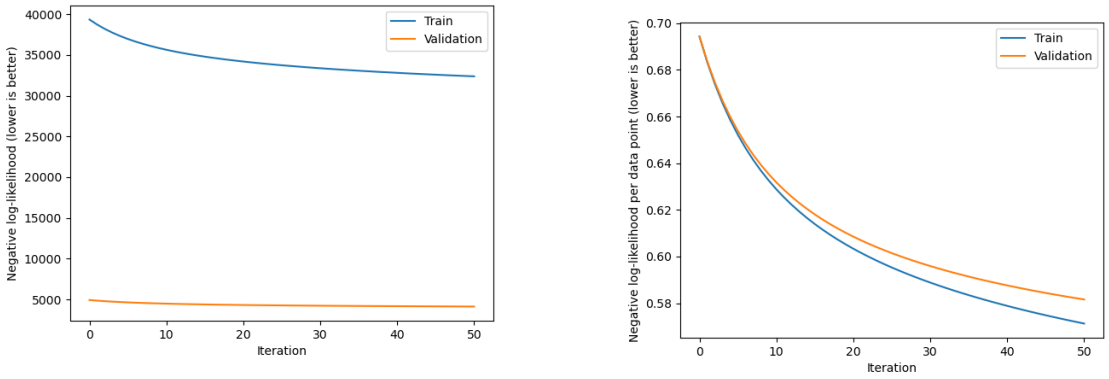


Figure 2: Training and validation accuracy over time.

c. The final training and validation accuracies after 50 iterations of training (see above) were 71.6% and 70.3% respectively[4].

[2] Since z is the sigmoid function, we have $\partial z/\partial \theta_i = z(1-z)$ and $\partial z/\partial \beta_j = -z(1-z)$.
[3] In each iteration, $\boldsymbol{\theta}$ is updated before $\boldsymbol{\beta}$.
[4] According to the Faculty of Arts and Science, this is equivalent to a B-



(a) Absolute negative log-likelihood. Note that the validation set, being significantly smaller, has a much smaller negative log-likelihood.

(b) Negative log-likelihood, normalized by number of data-points.

Figure 3: Negative training and validation log-likelihood over time (lower is better).

d. See Figure 4. Note the S-shape of the curve resulting from the use of the sigmoid function with respect to a linear variation over its parameter $\theta - \beta_j$. These plots may be interpreted as the predicted probability of success on a question with a measure of difficulty corresponding to β_j with respect to student competency measured by θ.

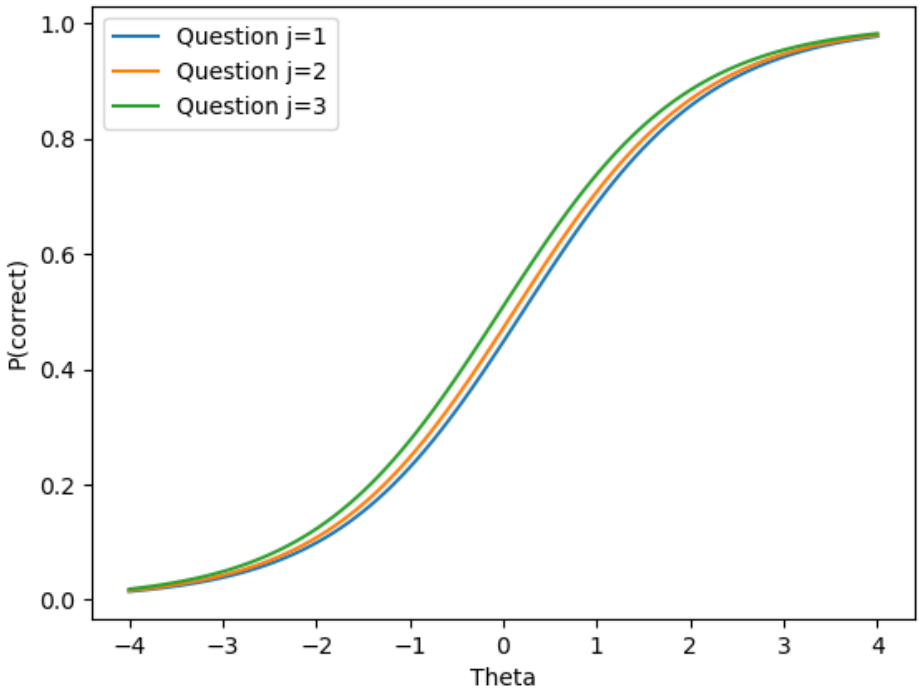


Figure 4: $p(\text{correct} \mid \theta, \beta_j) = \dfrac{\exp(\theta - \beta_j)}{1 + \exp(\theta - \beta_j)}$ with respect to θ for $j \in \{1, 2, 3\}$.

3. **Option 2: Neural Networks**

a. The differences between alternating least squares (ALS) and neural networks are as follows:

i. Firstly, the purpose of ALS is to reduce the high-dimensional input data into two lower-dimensional factor matrices meant to represent questions and students in the context of this problem. Neural networks, on the other hand, attempt to represent the entirety of the data as a set of latent features rather than a differentiating between questions and students.

ii. Secondly, ALS alternates[5] between optimizing each of the two latent feature matrices in order to factor the matrix, while neural networks primarily use the multivariate chain rule to calculate the gradient in order to backpropagate, thus updating all of the weights of the network at once.

ALS thus represents a more linear model optimized by an iterative process, whereas neural networks are nonlinear and more parallelizable.

iii. Thirdly, ALS is an unsupervised learning algorithm, while neural networks can be used for both supervised and unsupervised learning. In this sense, the scope of neural networks as a whole is much broader than that of ALS; neural networks can be used for a much wider set of tasks than simply dimensionality reduction.

b. The forward pass was implemented according to the docstring.

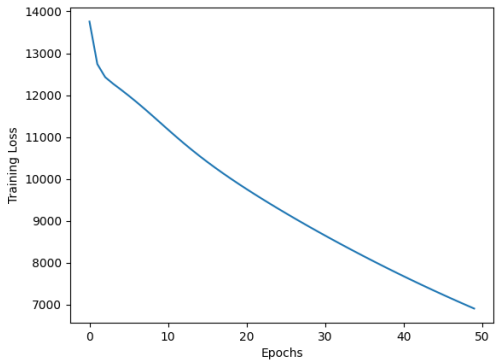
c. For every permutation of hyperparameters (α, n, k) of learning rate over

$$\begin{aligned} \text{Learning rate} \quad & \alpha \in \{0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\} \\ \text{Iteration count} \quad & n \in \{10, 25, 50, 100\} \\ \text{Model dimensionality} \quad & k \in \{10, 50, 100, 200, 500\} \end{aligned}$$

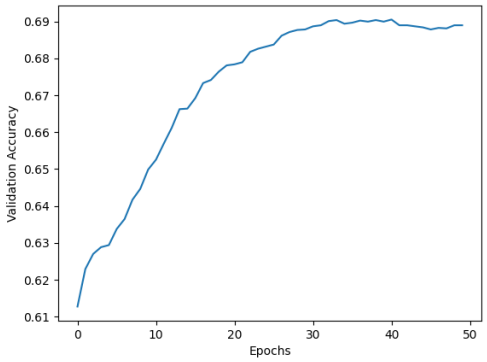
over iterations, $k^* = 50$ was able to most consistently yield the highest validation accuracy. The highest validation accuracy with $k^* = 50$ was with a learning rate of $\alpha = 0.01$ and $n = 50$ iterations, yielding a validation accuracy of 68.1%.

d. The final test accuracy was 68.5%. The training loss and validation accuracies over time are shown in Figure 5.

[5] Hence the name.



(a) Training loss over time without modified loss.

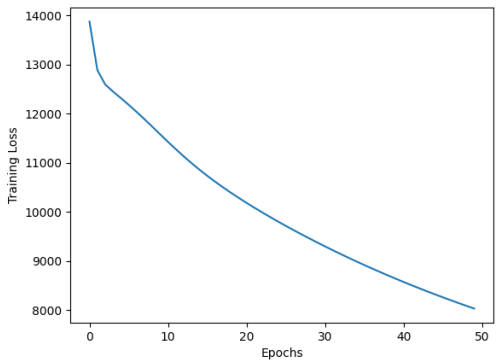


(b) Validation accuracy over time wihout modified loss.

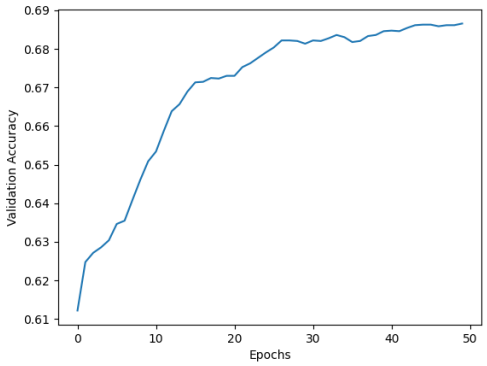
Figure 5: Training loss and Validation accuracy over time without modified loss (lower loss and higher accuracy is better).

e. For the regularization term $\lambda = 0.001$, the test accuracy was 68.4%, and the validation accuracy was 68.7%. The adjusted loss function achieved a higher validation accuracy than the original loss function, however, it had slightly lower test accuracy than the original test function.

The validation accuracy over time, as shown in 6, behaves more erratically than when using the unmodified loss function, while the training loss over time behaves similarly to that of the unmodified loss function, though due to the added term, in the loss function, is shifted upwards.



(a) Training loss over time with modified loss.



(b) Validation accuracy over time with modified loss.

Figure 6: Training loss and Validation accuracy over time with modified loss (lower loss and higher accuracy is better).

4. **Ensemble**

The ensemble prediction was obtained by composing 3 datasets, each the same size as the original dataset, from the original dataset with replacement. For each bootstrapped dataset, an IRT model was trained on that dataset. The IRT models, once trained, made predictions on the test and validation data; the three models' predicted probability to answer each question correctly was then averaged to

make the prediction. The final validation accuracy was 70.7%, and the final test accuracy was 70.1%.

While the test accuracy is lower than that of the non-aggregated IRT model, the validation accuracy was nominally higher. These changes in test and validation accuracy could be attributed to the random aspect of bootstrapping; the training datasets were obtained by sampling with replacement, which could have caused some of the datasets to be skewed towards particular values, despite the average distribution of the bootstrapped datasets being the same or extremely similar as that of the original dataset. However, these changes are relatively small and overall don't change the accuracy of the model in any meaningful ways.

Part B

Introduction

To attempt to improve on the results obtained in **Part A, Question 2–Item Response Theory**, we employ the use of a regularized *multidimensional item response theory* (MIRT) model. Our underlying hypothesis is that the relatively low complexity of the original IRT model (and low resulting training accuracy providing evidence for underfitting) suggests that the performance of model may be improved in general by increasing the number of parameters.

The model used is the one defined by Reckase (2009) and is a multidimensional generalization of the 2-parameter logistic (2PL) model, which is, in turn, a simplification of the 3-parameter logistic (3PL) first formulated by Birnbaum (1968).

The MIRT Model

A MIRT model with dimension m predicts the probability of a given student i answering a question j as

$$p \stackrel{\text{def}}{=} p(c_{ij} = 1 \mid \boldsymbol{\alpha}, \boldsymbol{\theta}, \mathbf{d}) = \frac{\exp(\boldsymbol{\theta}_i \cdot \boldsymbol{\alpha}_j + d_j)}{1 + \exp(\boldsymbol{\theta}_i \cdot \boldsymbol{\alpha}_j + d_j)} = \frac{\exp\left(\sum_{\ell=1}^{m} \theta_{i\ell}\alpha_{j\ell} + d_j\right)}{1 + \exp\left(\sum_{\ell=1}^{m} \theta_{i\ell}\alpha_{j\ell} + d_j\right)}$$

given parameters:

1. $\boldsymbol{\alpha}$: a $N_{questions} \times m$ matrix. By analogy to the original IRT model, the purpose of each row $\boldsymbol{\alpha}_\ell$ is comparable to that of $\boldsymbol{\beta}$ in the original model. In other words, in our MIRT model, each row is a latent representation of each question, whereas in our original model, each element of $\boldsymbol{\beta}$ is a scalar representation of each question (analogous to its difficulty).
2. $\boldsymbol{\theta}$: a $N_{students} \times m$ matrix. By analogy to the original IRT model, the purpose of each row $\boldsymbol{\theta}_\ell$ is comparable to that of $\boldsymbol{\theta}$ in the original model. In other words, in our MIRT model, each row is a latent representation of each student, whereas in our original model, each element of $\boldsymbol{\theta}$ is a scalar representation of each student (analogous to their competence).
3. $\mathbf{d}$: a vector of size $N_{questions}$. Essentially a bias term (referred to as an "intercept term" by Reckase (2009)).

The dimension hyperparameter m of the model is proportional to the complexity and number of parameters of the model. As will be discussed in more detail later, this value must be chosen carefully as to balance both the capability of the model to fit to the training data and the ability of the model to properly generalize.

Model Optimization

Similar to the original model, the log-likelihood function of the model is given by

$$\ell \stackrel{\text{def}}{=} \sum_i \sum_j c_{ij} \log(p) + (1 - c_{ij}) \log(1 - p).$$

Optimization of this model is performed by iteratively applying the update rule

$$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + \gamma \frac{\partial \mathcal{J}}{\partial \boldsymbol{\alpha}} \quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \gamma \frac{\partial \mathcal{J}}{\partial \boldsymbol{\theta}} \quad \mathbf{d} \leftarrow \mathbf{d} + \gamma \frac{\partial \mathcal{J}}{\partial \mathbf{d}}$$

for a learning rate γ and fitness function

$$\mathcal{J} = \ell - \lambda\mathcal{R}^{[6]}$$

where $\mathcal{R} = \|\boldsymbol{\alpha}\|^2 + \|\boldsymbol{\theta}\|^2 + \|\mathbf{d}\|^2$ is an L2 regularization term used to prevent overfitting (this will be addressed in more detail later).

Model Visualization

In the context of the original model, Figure 4 shows the resulting probability of correctness with respect to the parameter θ_i for some fixed β_j.

In the context of the MIRT model, Figures 7 and 8 visualize the contour of $p(c_{ij} = 1 \mid \boldsymbol{\alpha}_i, \boldsymbol{\theta}_i, \mathbf{d})$ with respect to $\boldsymbol{\theta}_i \in \mathbb{R}^2$ for some fixed $\boldsymbol{\alpha}_j$. In some sense, $\boldsymbol{\alpha}_j$ acts as to linearly transform an m-dimensional sigmoid function p in the space over $\boldsymbol{\theta}_j \in \mathbb{R}^m$, and vice versa. Intuitive interpretation of this visualization in the context of the student/question context is difficult, but these plots should be sufficient to demonstrate that the overall "shape" of the model's output with respect to its parameters is only extended to higher dimensions rather than being fundamentally altered.

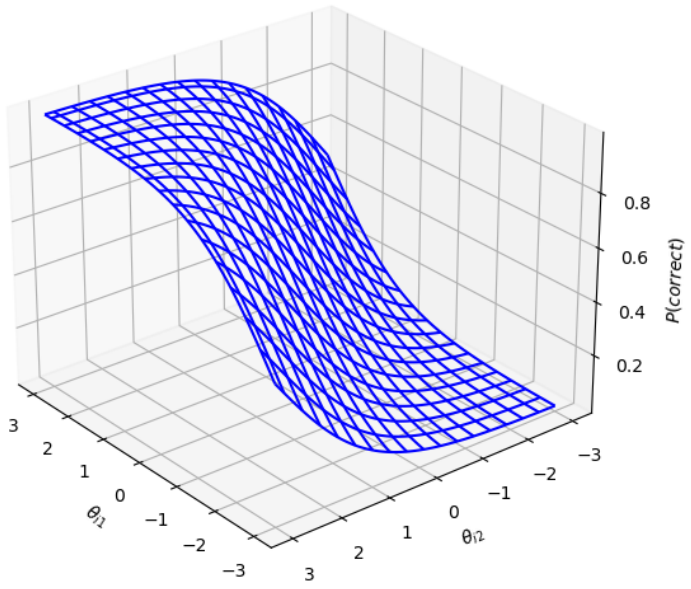


Figure 7: $p(c_{ij} = 1 \mid \boldsymbol{\alpha}_j, \boldsymbol{\theta}_i, \mathbf{d})$ with respect to $\boldsymbol{\theta}_i$. Note that $\boldsymbol{\alpha}$ has been fixed to $\langle 1, 1\rangle$ for exemplary purposes.

[6] Note that we subtract the regularization term from the log-likelihood term. Since we are looking to *maximize* the log-likelihood, penalization should be done by *subtracting* a quantity proportional to the complexity of the model.

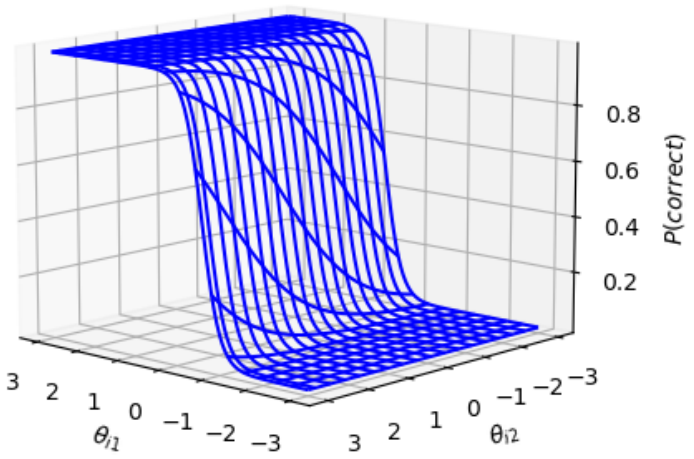


Figure 8: $p(c_{ij} = 1 \mid \boldsymbol{\alpha}_j, \boldsymbol{\theta}_i, \mathbf{d})$ with respect to $\boldsymbol{\theta}_i$. Note that $\boldsymbol{\alpha}$ has been fixed to $\langle 5, 1 \rangle$ for exemplary purposes.

Methodology

We trained the model for 50 iterations with a learning rate $\gamma = 0.02$. The model was initialized with dimensionality $m = 3$ with a regularization weight $\lambda = 2$. The hyperparameters m and λ were chosen via an automatic hyperparameter selection process wherein 100 pairs (m, λ) were generated from a uniform distribution where

$$m \in \{2, 3, \ldots, 6\} \quad \lambda \in [0, 4],$$

and the corresponding final validation accuracy $\mathcal{A}(m, \lambda)$ was evaluated after 50 training iterations as means of approximation for

$$\arg\max_{(m,\lambda)} \mathcal{A}(m, \lambda).$$

The final testing and validation accuracies were found to vary by $< 1\%$ between runs.

Results and Comparison

Figure 9 shows the accuracy obtained by the model over time (i.e. with respect to number of iterations). The final validation accuracy was 69.7%, and the final test accuracy was 69.6%.

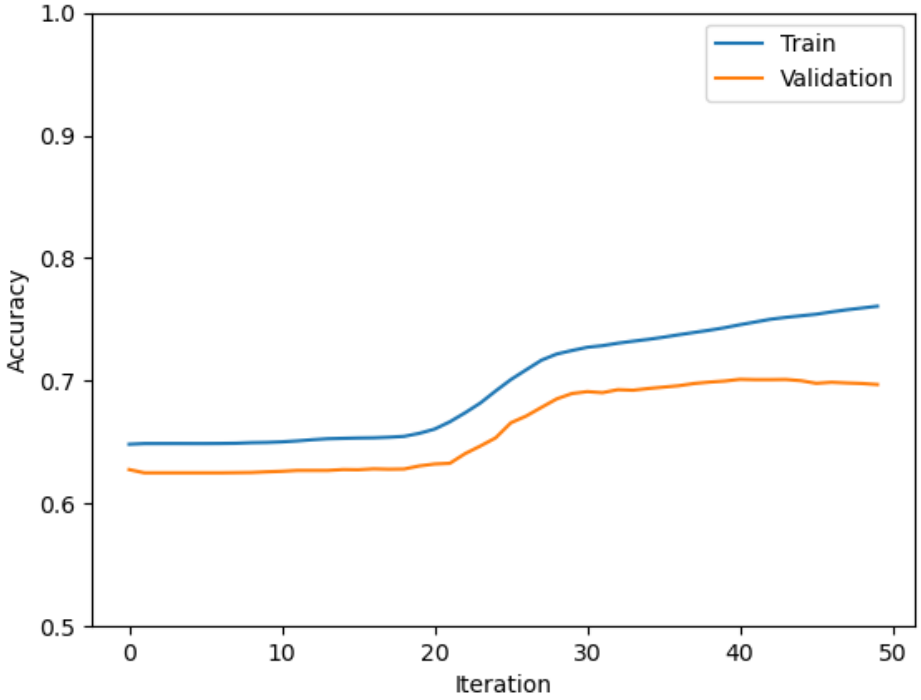


Figure 9: MIRT training and validation accuracy over time.

Model	Training Accuracy	Validation Accuracy
MIRT ($m = 3, \lambda = 2$)	76.1%	70.4%
MIRT ($m = 3, \lambda = 0$)	78.0%	69.4%
MIRT ($m = 10, \lambda = 0$)	88.6%	69.5%
IRT (original)	71.6%	70.3%

Table 2: Results comparing the *highest* (not final) training and validation accuracies obtained during training across different model configurations. MIRT ($m = 3, \lambda = 2$) is the model configuration chosen for demonstration purposes.

See Table 2. Note the slightly lower resulting validation accuracy (and increased training accuracy) for the model configuration which omits the regularization term ($m = 3, \lambda = 0$) when compared to the demonstration model. These results motivate the use of a regularization term to mitigate some of the effects of overfitting, likely from the increased model complexity. By taking this hypothesis to the extreme for the configuration ($m = 10, \lambda = 0$), we see a dramatically improved training accuracy. While the *highest* validation accuracy remains competitive with the other model configurations, Figure 10 shows as the model begins to improve with respect to its training accuracy, validation accuracy *decreases* to 66.2%, implying that the model has begun to overfit.

Model Limitations

The results shown in Table 2 implies that the MIRT model, with its arbitrary complexity and capability of fitting to the training data, is still limited in its ability to generalize. Although further analysis is warranted, we propose two immediate hypotheses:

1. The model's learned latent representations $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ of information about the questions and students lack in generalizability. In other words, these parameter vectors encode too much information about the training data points themselves rather than providing a general representation of their corresponding subjects.

   This problem may potentially be addressed by changing the regularization metric used; the L2 regularization metric used for the purposes of demonstration is a fairly naïve metric, and problem-specific information could be applied to optimize the regularization method further.

2. The increased number of parameters in the model necessitates more training data. In other words, the ratio of training data points to $N_{students} \times N_{items}$ is too small for the purposes of sufficiently training this model. A case for this hypothesis is made by considering the much smaller difference between the training and validation accuracies of the smaller, original IRT model.

   While the amount of training data provided is non-negotiable, this hypothesis may be tested by artificially omitting training data to see the resulting impact on validation accuracy. In other words, given a model $\mathbf{M}$ (i.e. the MIRT model or the original IRT model), we may define a function

   $$\mathcal{A}(p \mid \mathbf{M})$$

   describing the resulting training accuracy as a function of a proportion $0 \leq p \leq 1$ limiting the amount of original training data for the purposes of training.

   If we see that as p approaches 1, we do not see a significant improvement in $\mathcal{A}$, then by extrapolation[7], we may assume that the model $\mathbf{M}$ is not limited by the amount of training data provided, rather, the model is being limited by some other factor inherent to the model itself (see hypothesis 1).

Conclusions

As shown in Table 2, we see that the MIRT model is essentially equivalent to the original model in terms of validation accuracy. Despite this, the pattern of consistently higher training accuracies for the MIRT model when compared to the original model supports the original hypothesis that model complexity (specifically the model's dimesnion m) is proportional to the model's capability to fit the training data.

With respect to the data given, the modifications to this model do not provide clear improvements when evaluated over the datasets on which the model was not directly trained. Despite this, both the flexible complexity of the model and demonstrably improved capability of fitting to the training data provide a compelling case for MIRT as a more robust model compared to the original model in applications where a greater relative quantity of training data is available.

[7] Specifically, we would attempt to use this data to predict $\mathcal{A}(p)$ as p increases beyond 1 to determine whether the model might benefit from additional training data, if it were somehow obtained.

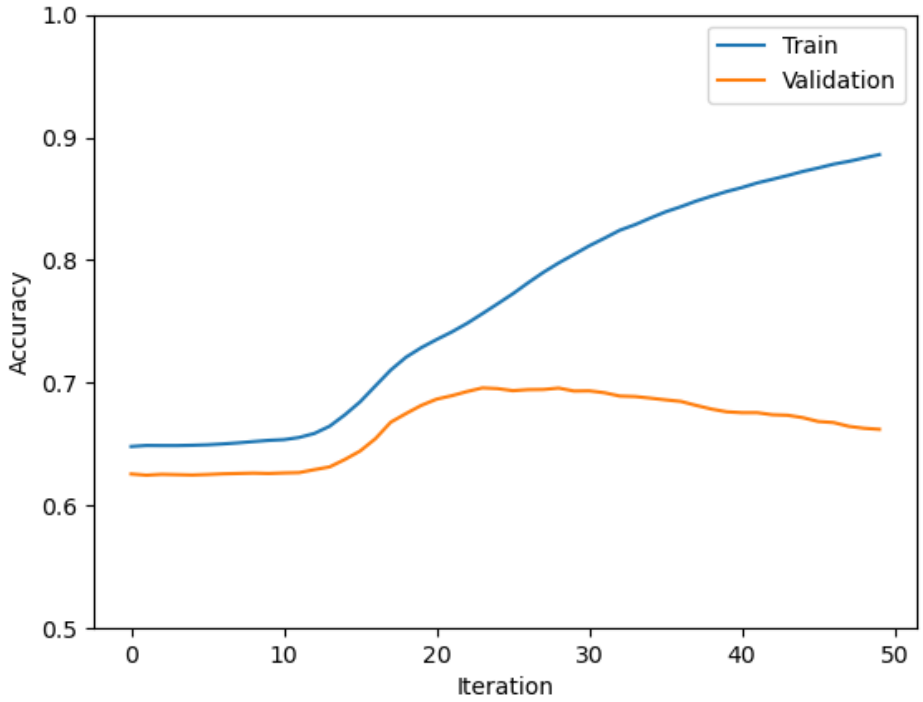


Figure 10: MIRT training and validation accuracy over time for $(m = 10, \lambda = 0)$, an extreme example of high model dimensionality demonstrating the capability of the model to better fit the provided data, generalizability notwithstanding.

Individual Contributions

The work on this project was distributed between its authors as follows:

- **Ian Huang**
  - Part A: Questions 1 and 2.
  - Part B: research and analysis.
  - Typesetting and editing.
- **Benjamin Liu**
  - Part A: Questions 3 and 4.
  - Part B: hyperparameter tuning, data visualization.
  - Derivations, gradient checking.

References

A. Birnbaum. Some latent trait models and their use in inferring an examinee's ability. *Statistical Theories of Mental Test Scores*, 1968.

M. Reckase. *Multidimensional item response theory*. Springer, 2009.