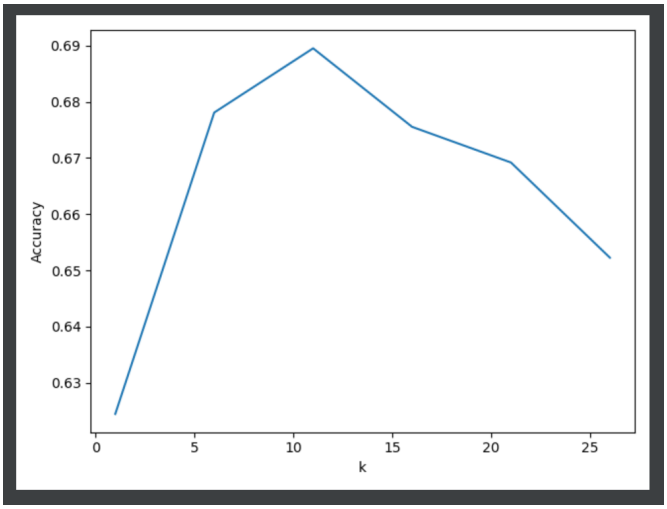
CSC311 Final Project

Ziqi Zhu[1006172204]
Qiming Ye[1005432373]
Yifan Qu[1005354894]

Part A

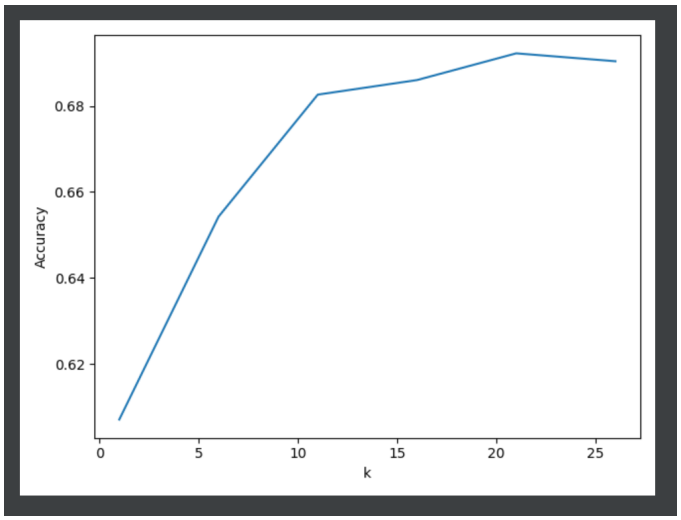
4.1 k-Nearest Neighbor. (Qiming Ye)

(a) different values of k are {1, 6, 11, 16, 21, 26}
The accuracy of the validation data as a function of k is shown on the graph below.



(b) different values of k are {1, 6, 11, 16, 21, 26}
Since when k* = 11, k has the highest performance on validation data. So we choose k* = 11 and the validation accuracy is 0.6895286480383855; the final test accuracy is 0.6841659610499576 (as graph shown below)

(c) different values of k are {1, 6, 11, 16, 21, 26}
Since when k* = 21, k is optimal. So we choose k* = 21, and the validation accuracy is 0.69037538808919; the final test accuracy is 0.6816257408975445



```
Sparse matrix:
[[nan nan nan ... nan nan nan]
 [nan  0. nan ... nan nan nan]
 [nan nan  1. ... nan nan nan]
 ...
 [nan nan nan ... nan nan nan]
 [nan nan nan ... nan nan nan]
 [nan nan nan ... nan nan nan]]
Shape of sparse matrix:
(542, 1774)
Validation Accuracy: 0.6244707874682472
Validation Accuracy: 0.6780976573525261
Validation Accuracy: 0.6895286480383855
Validation Accuracy: 0.6755574372001129
Validation Accuracy: 0.6692068868190799
Validation Accuracy: 0.6522720858029918
Validation Accuracy: 0.6841659610499576
Test accuracy with k star is 0.6841659610499576
Validation Accuracy: 0.607112616426757
Validation Accuracy: 0.6542478125882021
Validation Accuracy: 0.6826136042901496
Validation Accuracy: 0.6860005644933672
Validation Accuracy: 0.6922099915325995
Validation Accuracy: 0.69037538808919
Validation Accuracy: 0.6816257408975445
Test accuracy with k star is 0.6816257408975445
```

graph for 4.1(a)(b)(c)

(d) Since from the results of (a) ~ (c), we can have the conclusion that:
User-based data on validation accuracy is worse than item-based data on validation accuracy.
The user-based data on final test accuracy is better than item-based data on final test accuracy.
Also, the calculating time on user-based data consumes less than item-based data.
So, user-based collaborative filtering performs better than item-based collaborative filtering.

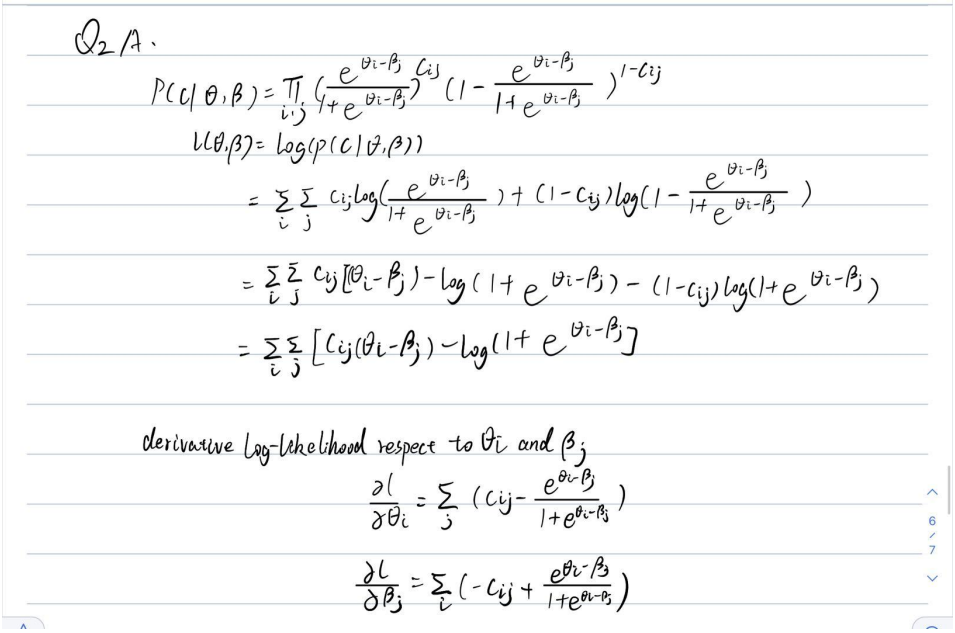
(e)

1) The calculating time is long. Although the user-based needs less time compared to item-based, it is still time-consuming.

2) The entire data is so large that it is hard to find the relationships since the model is a non-parametric model, which means it cannot be parameterized.

4.2 Item Response Theory (Ziqi Zhu, Yifan Qu)

(a)



Q2.4.

$$p(C|\theta,\beta)=\prod_{i,j}\left(\frac{e^{\theta_i-\beta_j}}{1+e^{\theta_i-\beta_j}}\right)^{C_{ij}}\left(1-\frac{e^{\theta_i-\beta_j}}{1+e^{\theta_i-\beta_j}}\right)^{1-C_{ij}}$$

$$l(\theta,\beta)=\log(p(C|\theta,\beta))$$

$$=\sum_i\sum_j C_{ij}\log\left(\frac{e^{\theta_i-\beta_j}}{1+e^{\theta_i-\beta_j}}\right)+(1-C_{ij})\log\left(1-\frac{e^{\theta_i-\beta_j}}{1+e^{\theta_i-\beta_j}}\right)$$

$$=\sum_i\sum_j C_{ij}[(\theta_i-\beta_j)-\log(1+e^{\theta_i-\beta_j})-(1-C_{ij})\log(1+e^{\theta_i-\beta_j})$$

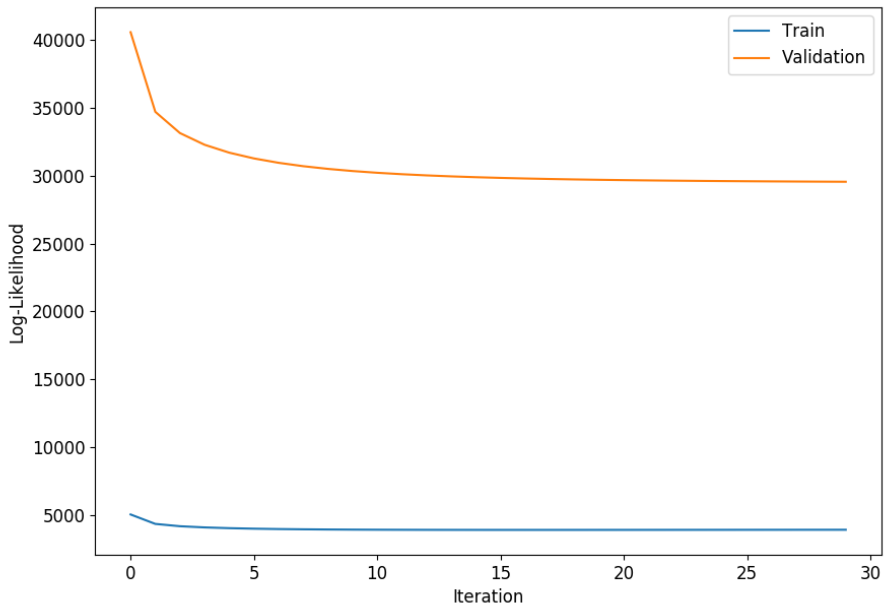
$$=\sum_i\sum_j[C_{ij}(\theta_i-\beta_j)-\log(1+e^{\theta_i-\beta_j}]$$

derivative log-likelihood respect to θ_i and β_j

$$\frac{\partial l}{\partial\theta_i}=\sum_j\left(C_{ij}-\frac{e^{\theta_i-\beta_j}}{1+e^{\theta_i-\beta_j}}\right)$$

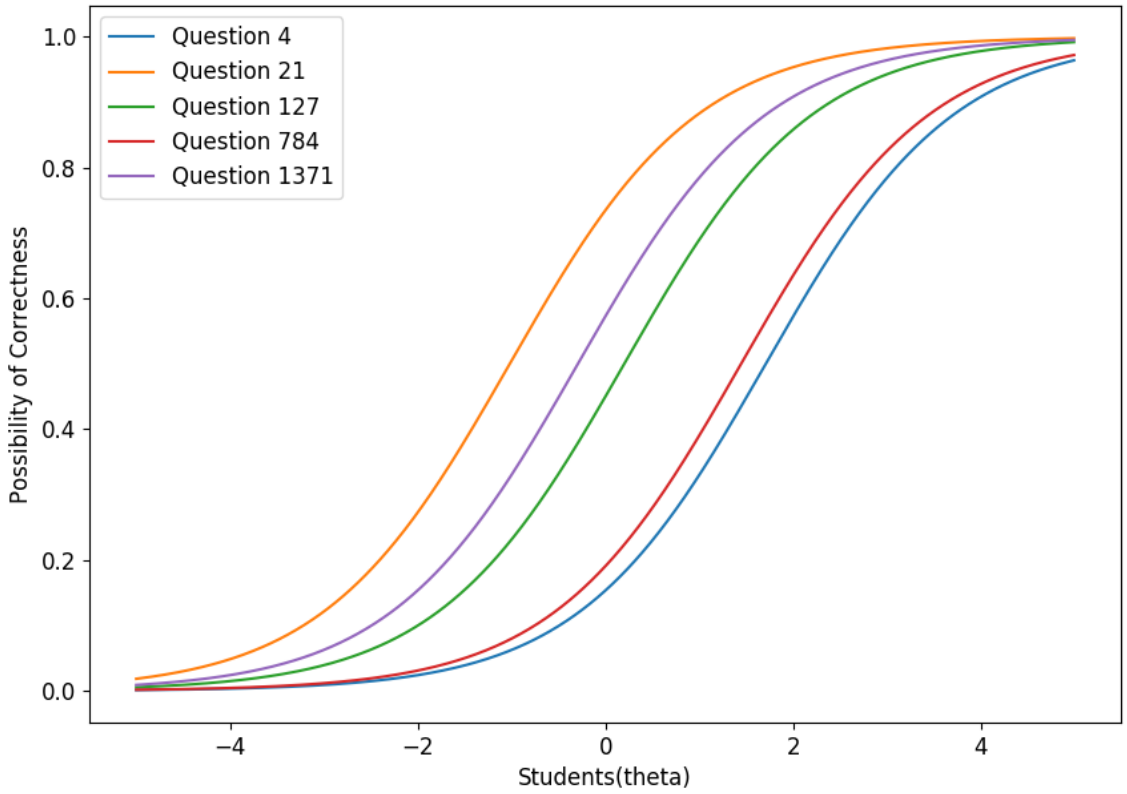
$$\frac{\partial l}{\partial\beta_j}=\sum_i\left(-C_{ij}+\frac{e^{\theta_i-\beta_j}}{1+e^{\theta_i-\beta_j}}\right)$$

(b) The chosen parameters are lr=0.02 and iteration=30, and the train- validation log-likelihood is shown as below



(c) Final Train Accuracy: 0.737, Validation Accuracy: 0.708

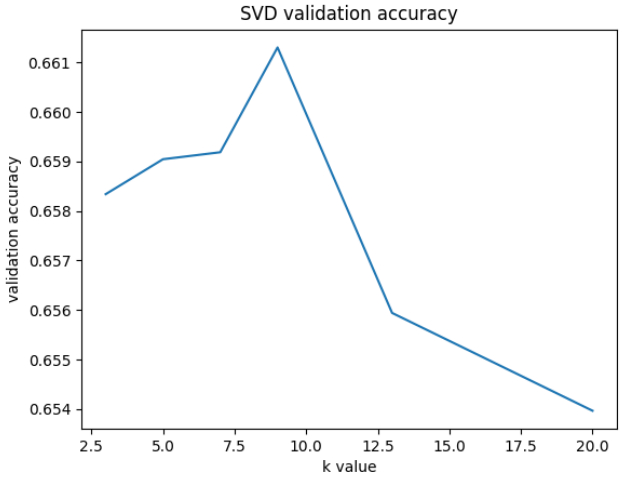
(d) The five questions we chose are 4, 21, 127, 784, and 1371. The plot of the probability of the correct response as a function of θ given a question is as follows. All 5 curves in the graph are increasing and



converge to 1.0 as the student's ability increases. This indicates that students with higher ability are more likely to answer a question correctly which makes sense.

4.3 option1 Matrix Factorization (Ziqi Zhu)

(a)
As the result shows, the chosen k is 9 and has a validation accuracy of 0.6613 and test accuracy of 0.6586.



(b)
In the given task, we are replacing the missing value of whether a student gives the correct answer for a question with the mean value of the correctness of other students.
This could bring bias to the data matrix and make different choices for eigenvalues.

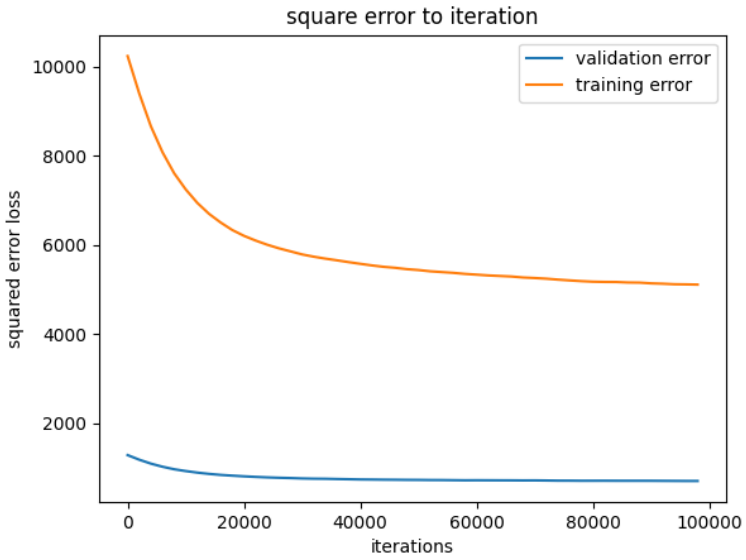
(c)
Please check the code in matrix_factorization.py.

(d)
The hyperparameters we are choosing are 0.05 learning rate and 100000 iterations. We tried k for [7, 9, 15, 25, 50, 100], the result is shown below

```
validation accuracy for k = 7: 0.69616144510302
validation accuracy for k = 9: 0.6953147050522156
validation accuracy for k = 15: 0.6900931414055885
validation accuracy for k = 25: 0.6933389782670054
validation accuracy for k = 50: 0.692492238216201
validation accuracy for k = 100: 0.6965848151284222
chosen k is 100, has validation accuracy 0.6965848151284222 and test accuracy 0.7011007620660458
```

When k=100 we have the highest validation accuracy, but it may be caused by the random updating on u and z.

(e)
The change of squared error when k=100 is shown below, as we can see the error is decreasing as the iteration increases, finally the squared error for training and validation data converges to some values. The final k=100 has a validation accuracy of 0.6966 and test accuracy of 0.7011.



4.4 Ensemble(Qiming Ye, Ziqi Zhu)

	Validation accuracy	Test accuracy
Ensemble	0.6778	0.6802
Original	0.6775	0.6754

Explain the ensemble process:

Steps:
1. Select the matrix factorization model developed in q3 as the base model.
2. Randomly select samples with replacement from the train_data until their size is the same as the original data.
3. Resampled data, perform the ALS algorithm 3 times with k=9, lr=0.05, and 50000 iterations as parameters, add the result to a matrix
4. Return the accuracy of the predictions for validation data and test data by using this matrix

5. Return the accuracy of the predictions for validation data and test data by using original train data who only perform the ALS algorithm once.

Do you obtain better performance using the ensemble? Why or why not?
Sometimes the performance of the ensemble is better, sometimes it is worse. Since during the selection process we randomly choose students and questions, some of the questions may be mispredicted as correct and sometimes may get wrong.

Part B Modified Item Response Theory

5.1 Formal Description(Yifan Qu)

In the original IRT model, βj represents the difficulty of question j, and θi represents the i-th student's ability. Then, the probability that the question j is correctly answered by student i is formulated as:

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

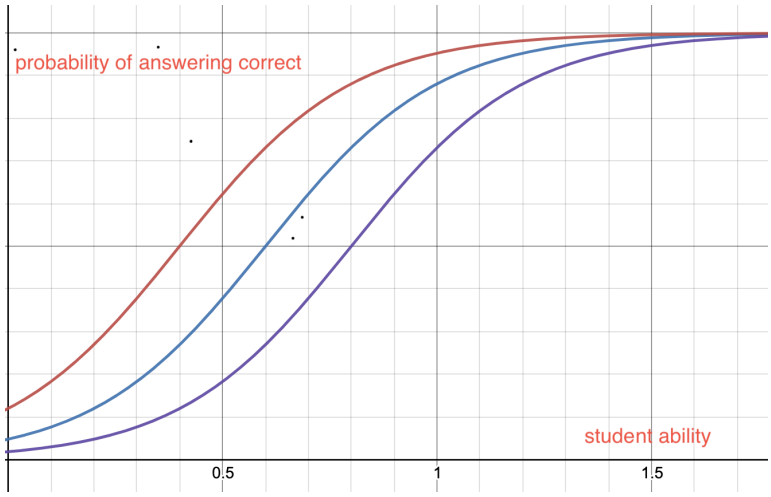
with the results: Train Accuracy: 0.737, Validation Accuracy: 0.708. We notice that both valid accuracy and train accuracy are not so high. It seems that the original model tends to be underfitting. Thus, we decide to add more parameters to better fit the model with data.

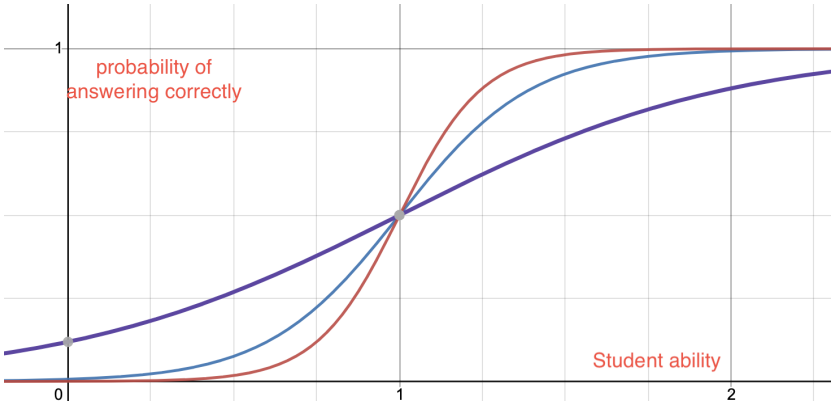
We find that there is another factor affecting the students' performance on some questions. Some questions could better distinguish students' performance than other questions.

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

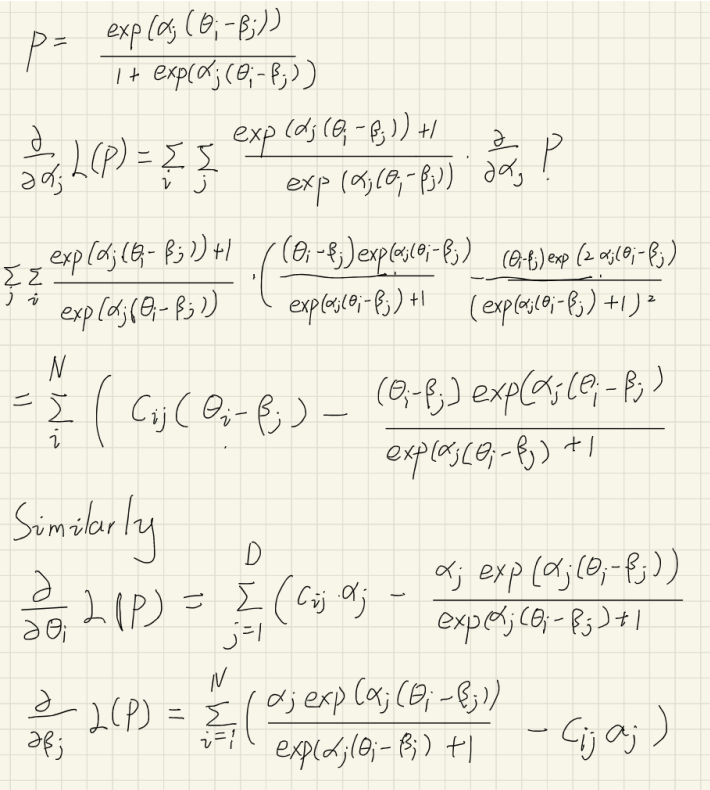
We also noticed that in the students' metadata, we are given the information of genders, birth and if a student is a premium_pupil. However other information is not as complete as gender, so we are interested in whether treating gender as a factor will improve the overall accuracy. Instead of assigning a random θ for each student, we are assigning θ to different genders.

5.2 Figure and diagram(Yifan Qu)





The top first image is the probability of correctness without alpha. We can see that without alpha terms, as a student's ability on this subject increases, they are more likely to get the correct answer in an even way. However, in the real world, for some questions, only after some knowledge threshold(i.e. student reaches some point) the problem could be solved. That scenario is shown in the second image. For those questions that could be solved only after having some ability, they have a high discrimination term(alpha term).

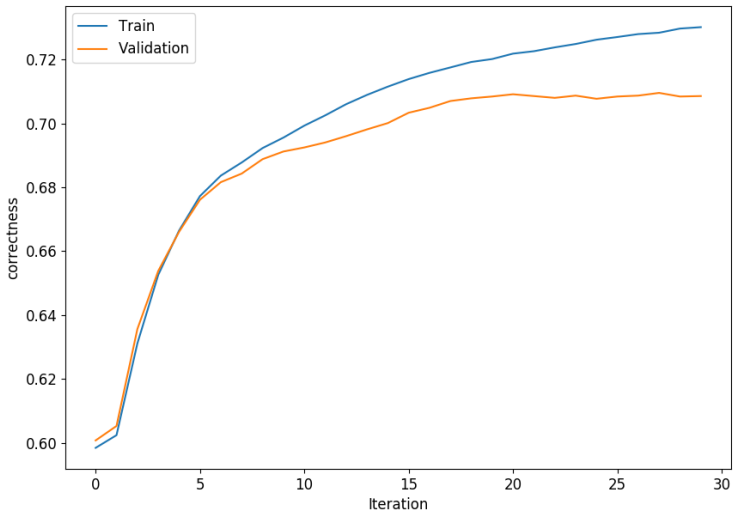


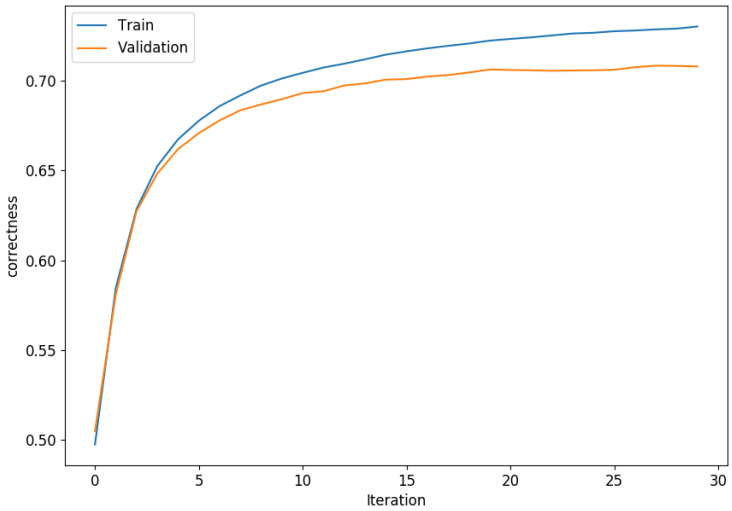
The page above shows how we get the gradient of the negative likelihood(loss function) by calculation, we would use that to update beta, theta and alpha.

5.3 Comparison or Demonstration(Yifan Qu)

Model Name	training accuracy	testing accuracy	loss
IRT	0.7387	0.708	29591
proposed IRT	0.739	0.71	30101
KNN	0.73	69.4	N/A

The proposed IRT has a better testing accuracy and a slightly higher loss compared to the baseline model.





As we can see, they are similar in training and validation data, while proposed irt has a slight advantage on validation rate.
first image is proposed
the second is original

5.4 Limitations(Qiming Ye, Ziqi Zhu, Yifan Qu)

Our approach will perform poorly when a student has a large variance on questions, the algorithm will have a wrong estimation and prediction on this student.

In our model, we have only used the alpha term on updating but not on evaluating since we set the threshold to be 0.5.

$sigmoid(\alpha x) = e^{\alpha x}/(1 + e^{\alpha x})$

From the equation above, we know that sigmoid is greater than 0.5 if x is greater than 0. And alpha is greater than 0 in our model, so it would not change the sign of x. Thus, has no effect on updating.
In future use, we might set a threshold to other values to get better performance.

Since in this dataset, the alpha term doesn't change a lot and is always greater than 0, so we haven't set any constraints on it. However, if there is another dataset that makes alpha change greatly and become negative, it might cause some undefined behaviour. Thus, to make our model more robust, we would process the alpha term with sigmoid or other activation functions.

We set two more parameters for our model, which increases the chance to be overfitting.
We also don't have sufficient data since the training data is not large enough.

To extend the model further, we would use subject metadata to classify discrimination terms for each question. Questions would have different discrimination rates based on their subject.
We can improve validation and test accuracy by using bagging aggregation since we can make variance smaller by using bagging, which helps our algorithm to reduce the wrong prediction.

Note:
Contributions are written in every title. Our three teammates discussed and solved most problems together.