Final Report

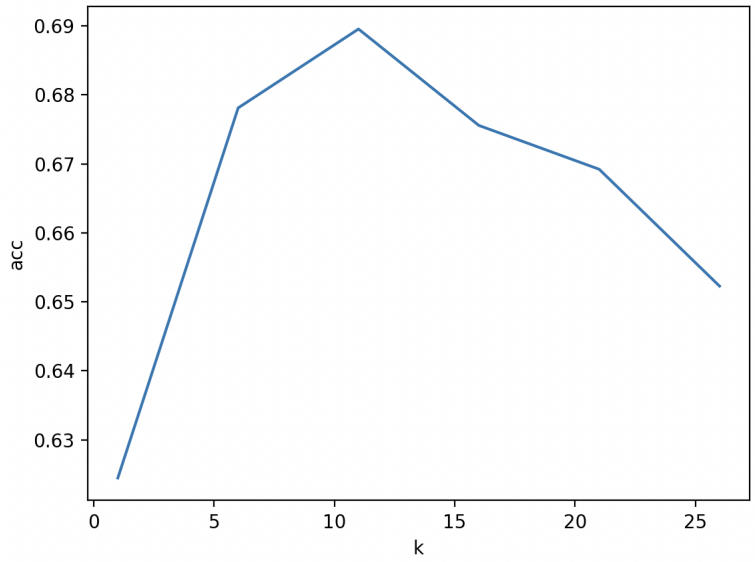
Xuerong Zhou, Qiyi Zhang, Zixiu Meng

March 31, 2023

1 Part A

1.1 Q1

(a) **User-based collaborative filtering**

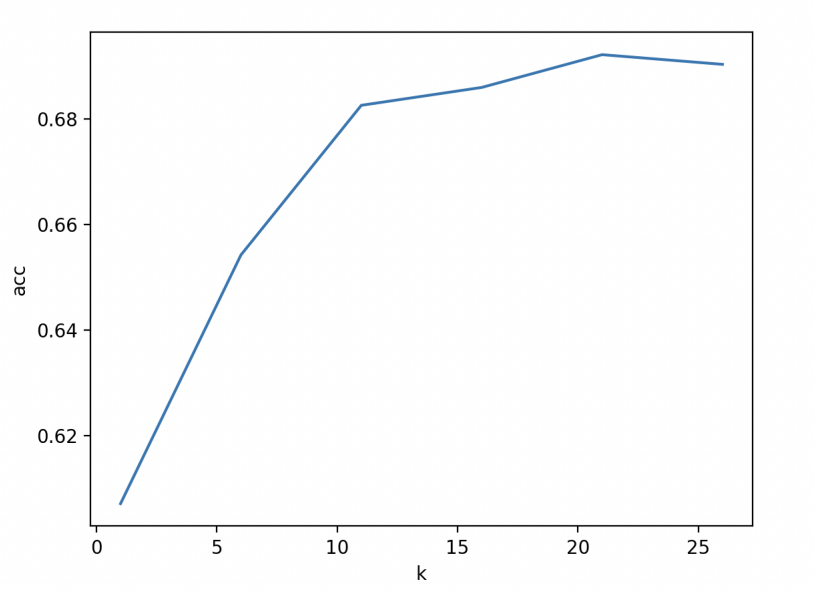



```
KNN impute by user
====================
when k: 1, Accuracy: 0.6244707874682472
when k: 6, Accuracy: 0.6780976573525261
when k: 11, Accuracy: 0.6895286480383855
when k: 16, Accuracy: 0.6755574372001129
when k: 21, Accuracy: 0.6692068868190799
when k: 26, Accuracy: 0.6522720858029918
====================
```

(b)

```
Highest performance will be achieved when k = 11,
and it has test accuracy 0.6841659610499576
```

(c) **Item-based collaborative filtering**:




```
KNN impute by Item
=====================
when k: 1, Accuracy on validation set: 0.607112616426757
when k: 6, Accuracy on validation set: 0.6542478125882021
when k: 11, Accuracy on validation set: 0.6826136042901496
when k: 16, Accuracy on validation set: 0.6860005644933672
when k: 21, Accuracy on validation set: 0.6922099915325995
when k: 26, Accuracy on validation set: 0.69037538808919
=====================
```

```
Highest performance will be achieved when k = 21,
and it has test accuracy 0.6816257408975445
```

Underlying assumption: If question A is answered the same way as question B is answered by other students, then for a specific student, his correctness on question A also match his correctness on question B.

(d) Answer: User-based collaborative filtering
Since the peak test accuracy for user-based collaborative filtering is slightly better than item-based collaborative filtering. And to achieve that accuracy, user-based collaborative filtering has k = 11, which is faster than item-based collaborative filtering with k=21.

(e)
1. Curse of dimensionality: That is, in high dimension, distance appears to be less meaningful because most points are approximately the same distance in high dimension. Therefore, KNN will be less effective in this case.
2. Computational complexity: KNN is time consuming and computationally expensive especially on a large data set and large k value. In this task, it is slower than other methods

1.2 Q2

(a) First, let $\theta = (\theta_1, \theta_2 ... \theta_d)$, where d represents the number of students. Let $\beta = (\beta_1, \beta_2 ... \beta_n)$ and n represents the number of question.

$$\begin{aligned}
L(C|\theta,\beta) &= p(C|\theta,\beta) \\
&= \prod_{i=1}^{d}\prod_{j=1}^{n}\left(\frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)}\right)^{c^{ij}}\left(1-\frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)}\right)^{1-c^{ij}} \\
&= \prod_{i=1}^{d}\prod_{j=1}^{n}\left(\frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)}\right)^{c^{ij}}\left(\frac{1}{1+exp(\theta_i-\beta_j)}\right)^{1-c^{ij}} \\
&= \prod_{i=1}^{d}\prod_{j=1}^{n}\left(\frac{exp(c_{ij}(\theta_i-\beta_j))}{1+exp(\theta_i-\beta_j)}\right)
\end{aligned}$$

Log-likelihood $\ell(\theta, \beta)$:

$$\begin{aligned}
logp(C|\theta,\beta) &= log\left(\prod_{i=1}^{d}\prod_{j=1}^{n}\left(\frac{exp(c_{ij}(\theta_i-\beta_j))}{1+exp(\theta_i-\beta_j)}\right)\right) \\
&= log\left(\prod_{i=1}^{d}\prod_{j=1}^{n}exp(c_{ij}(\theta_i-\beta_j))\right) - log\left(\prod_{i=1}^{d}\prod_{j=1}^{n}(1+exp(\theta_i-\beta_j))\right) \\
&= \sum_{i=1}^{d}\sum_{j=1}^{n}[(\theta_i-\beta_j)c_{ij} - log(1+exp(\theta_i-\beta_j))]
\end{aligned}$$

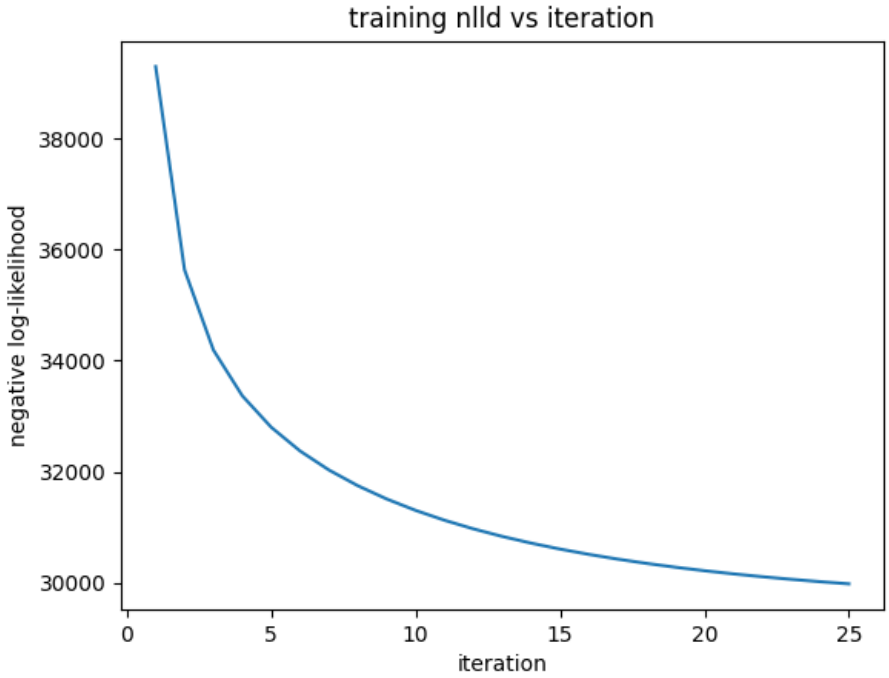
Derivative of log-likelihood w.r.t. θ_i:

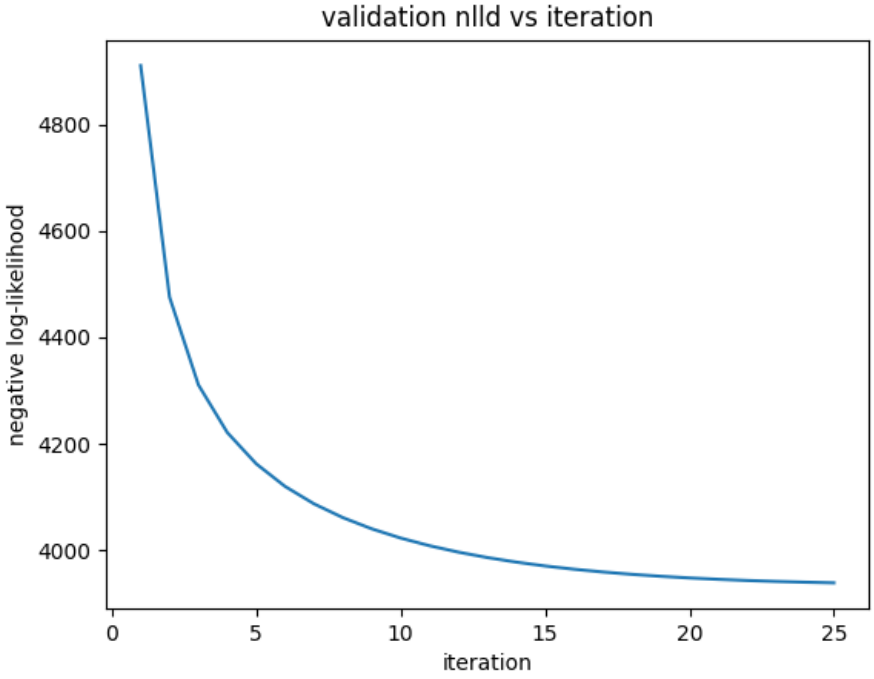
$$\begin{aligned}
\frac{\partial logp(C|\theta,\beta)}{\partial\theta_i} &= \frac{\partial}{\partial\theta_i}\sum_{i=1}^{d}\sum_{j=1}^{n}[(\theta_i-\beta_j)c_{ij} - log(1+exp(\theta_i-\beta_j))] \\
&= \sum_{j=1}^{n}\frac{\partial}{\partial\theta_i}\sum_{i=1}^{d}(c_{ij}(\theta_i-\beta_j) - log(1+exp(\theta_i-\beta_j))) \\
&= \sum_{j=1}^{n}\left(c_{ij} - \frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)}\right)
\end{aligned}$$

Derivative of log-likelihood w.r.t. β_j:

$$\begin{aligned}
\frac{\partial logp(C|\theta,\beta)}{\partial\beta_j} &= \frac{\partial}{\partial\beta_j}\sum_{i=1}^{d}\sum_{j=1}^{n}[(\theta_i-\beta_j)c_{ij} - log(1+exp(\theta_i-\beta_j))] \\
&= \sum_{i=1}^{d}\frac{\partial}{\partial\theta_i}\sum_{j=1}^{n}(c_{ij}(\theta_i-\beta_j) - log(1+exp(\theta_i-\beta_j))) \\
&= \sum_{i=1}^{d}\left(-c_{ij} + \frac{exp(\theta_i-\beta_j)}{1+exp(\theta_i-\beta_j)}\right)
\end{aligned}$$

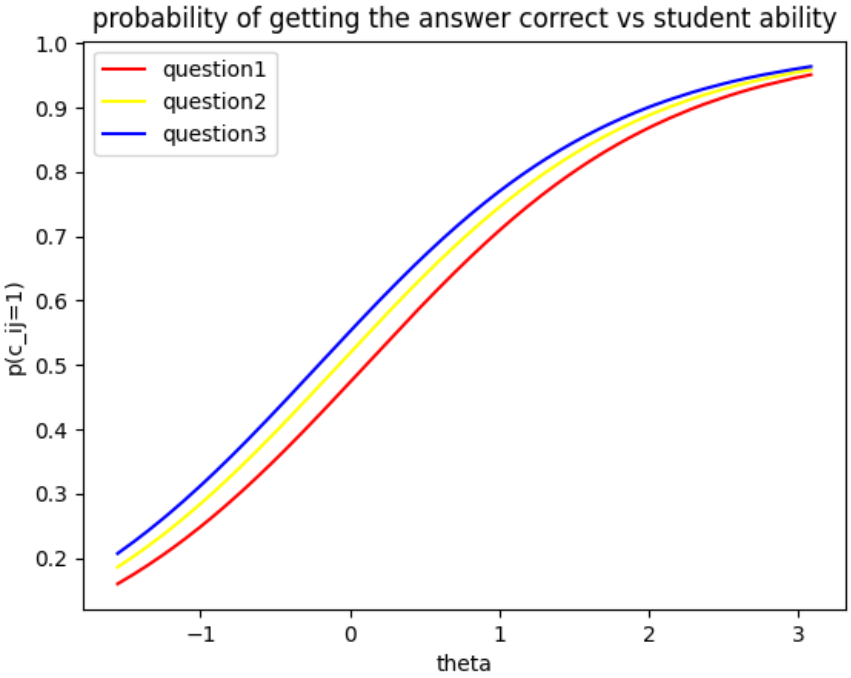
(b) For hyperparameters, we choose lr = 0.01 and iteration = 25.





(c) The final validation accuracy is 0.7070279424216765, the test accuracy is 0.7053344623200677.

(d) Let $j_1 = 1$, $j_2 = 10$, $j_3 = 100$. So j1 is the second question, j2 is the 11st question and j3 is the 101st question.



The graph has a sigmoidal shape, similar to the sigmoid function. Each curve on the graph represents the probability of a student correctly answering a given question. The curve's upward slope indicates a positive correlation between a student's ability and their likelihood of answering the question correctly.

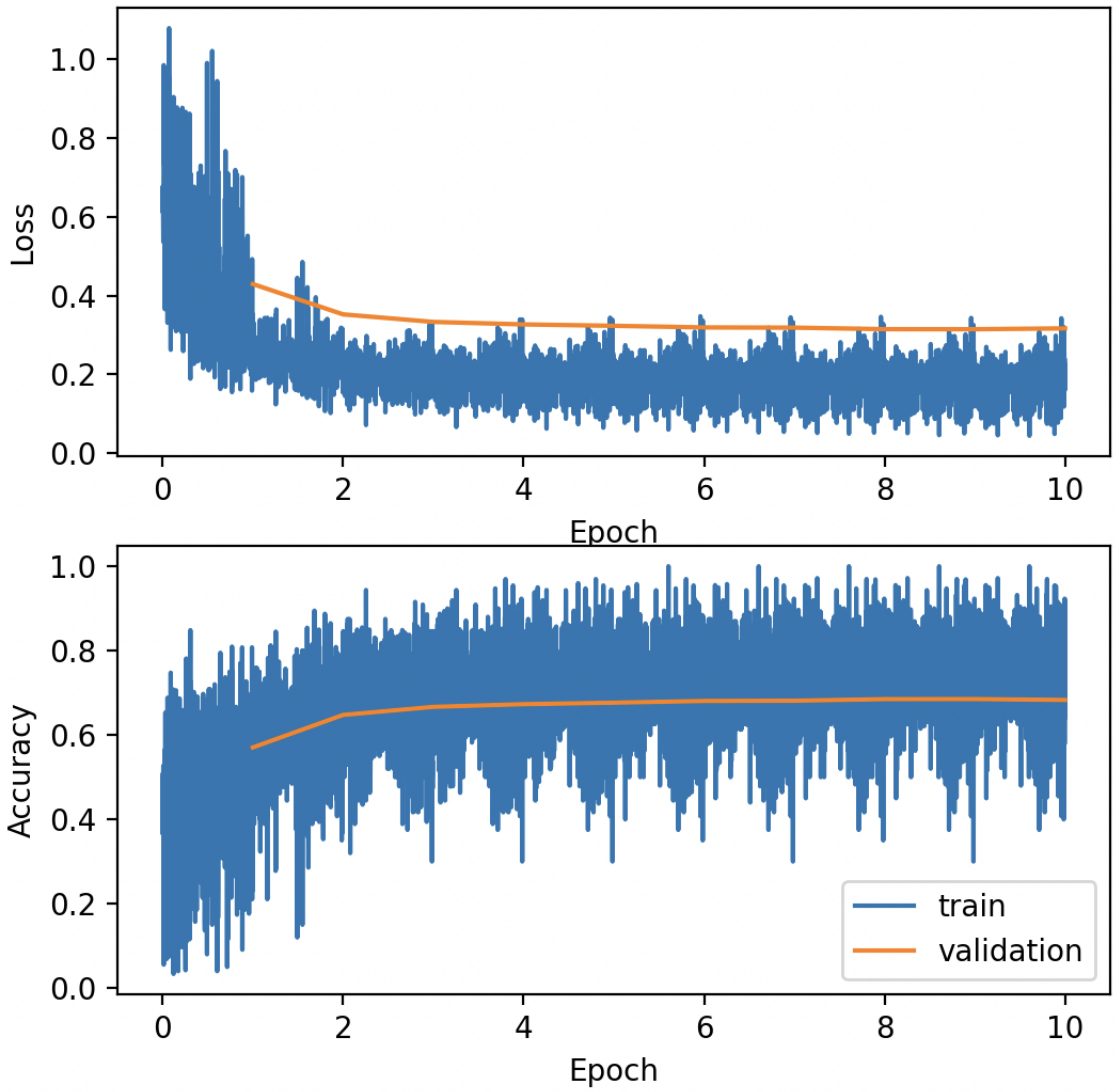
1.3 Q3

(a) ALS is different from neutral network and there are three out of all differences between them:

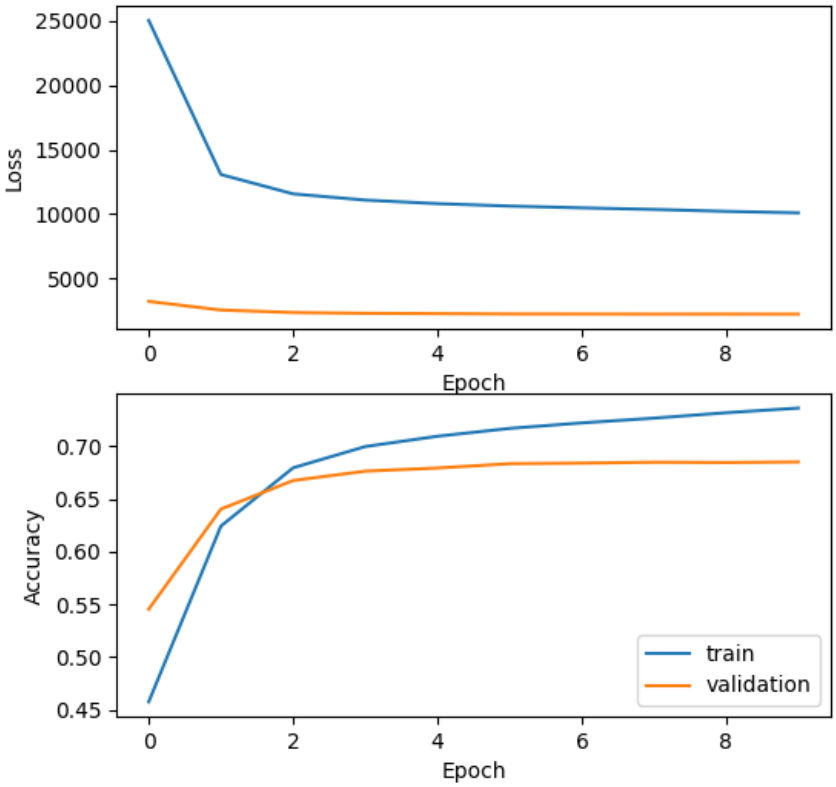
- a. neutral network might require multiple layers from the original input, whereas ALS just requires one layer.
- b. neutral network usually requires an activation function to compress the output from a linear regression in a range of $(-\infty, \infty)$ to a range of $[0, 1]$. However, there is no such operation for ALS usually.
- c. neutral network just has one weight vector parameter to optimize, whereas ALS generally have 2 parameters to optimize that are user vectors and movie feature vectors respectively. Note that the optimal parameters is the one that minimizes the distance between the target and the prediction reliant on that parameter.

(c) $k^* = 10$.

(d)

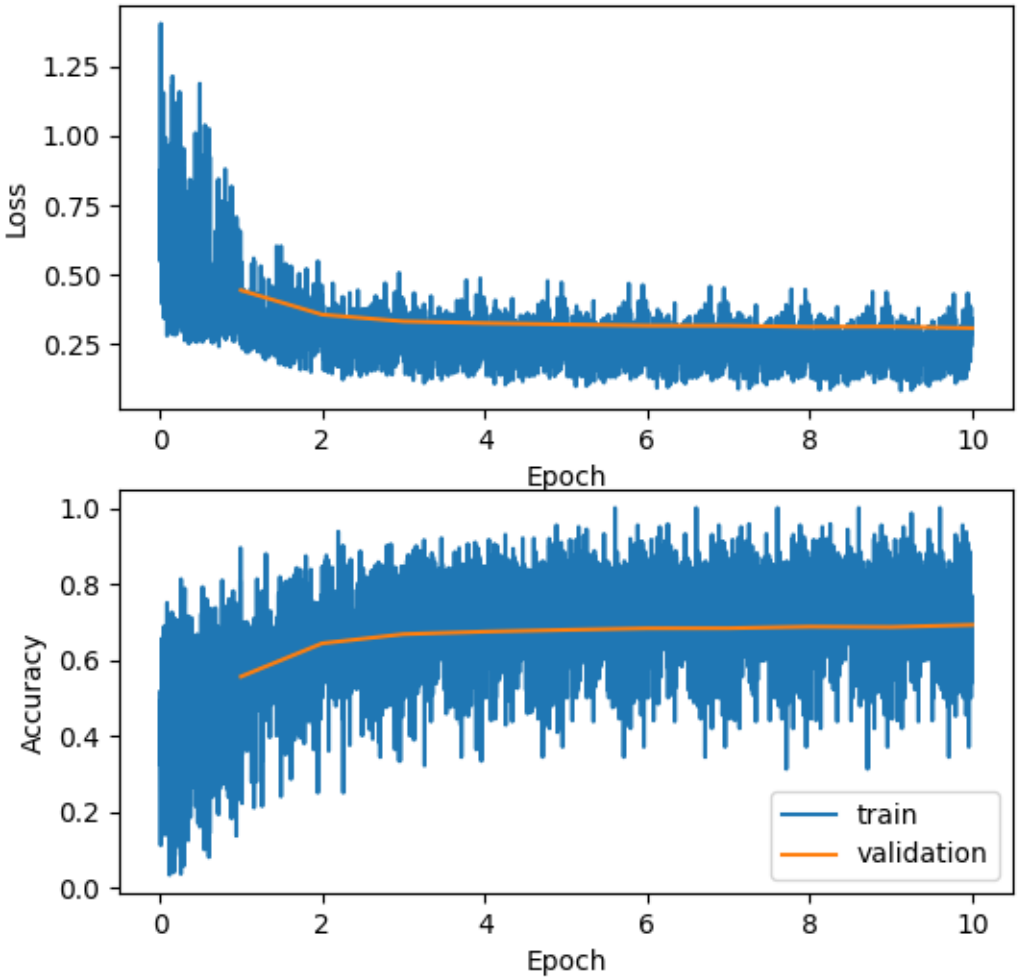


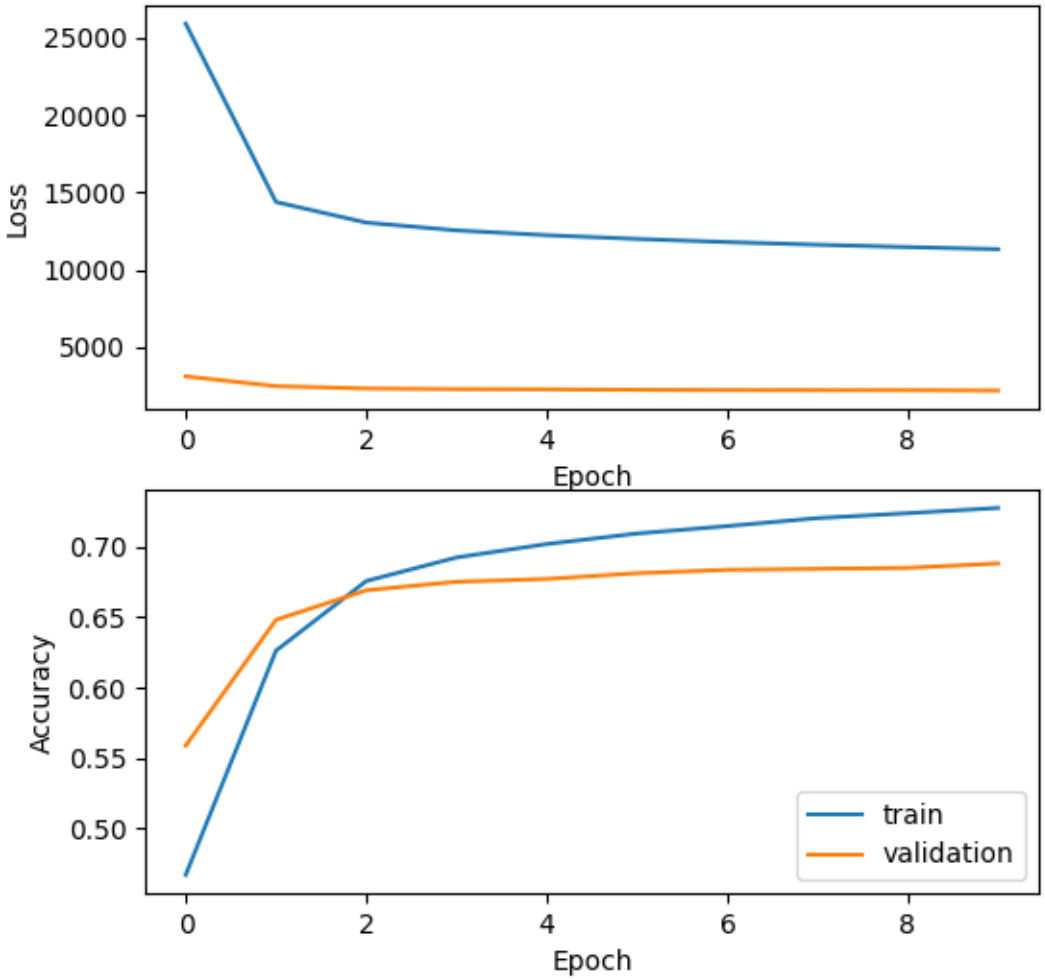
The test accuracy is 0.6802145074795372. Please note that we choose epoch as 10, learning rate as 0.005, k as 10 since this choice combination gives the best validation accuracy among our various choices. In addition, we want to compare whether the model is overfitting or underfitting, so we plot validation set and training set together. Then we use the average loss instead of the total loss to avoid the influence of data size. Furthermore, we also provide a total loss comparison and an average accuracy comparison between the training dataset and the validation dataset which presents the same trend in changing as epoch increases. Note that the graph indicates an early stopping point near epoch of 2 to avoid overfitting of the model.



(e) The optimal λ equals 0.01. With this optimal λ, we get the validation accuracy of 0.6874117979113745 and the test accuracy of 0.6892464013547841.

The model with the regularization performs better in this case since first, the accuracy is improved a bit and second, the model is more fitting. Please note that the model without regularization is a bit overfitting regarding loss, whereas the model with the regularization is fitting regarding both loss and accuracy. The below graphs show the comparison between the training and validation loss and accuracy.


Loss
1.25
1.00
0.75
0.50
0.25
0
2
4
6
8
10
Epoch
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
train
validation
0
2
4
6
8
10
Epoch



1.4 Q4

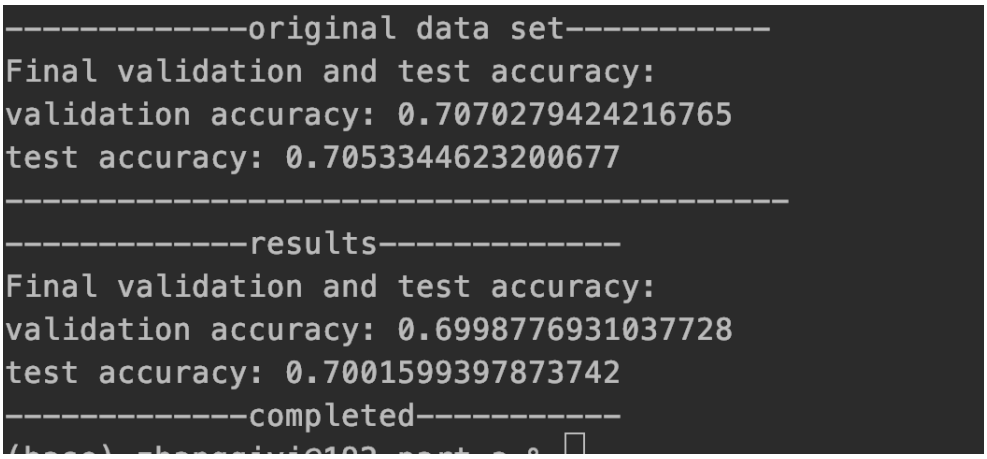
Ensemble process:

1. Firstly we bootstrapped train data from .csv file into 3 bagging bootstrapped data sets.

2. Then we trained 3 different IRT models from item_response.py using each of bootstrapped data set in step 1.

3. We evaluate accuracy of each model using theta and beta obtained from IRS model with validation data set and test data set

4. Lastly, we compute the average of accuracies from 3 IRS models to obtain the final validation and test accuracy.



From the result obtained from the ensemble model, we notice that compared to the IRS model trained using the original data set, the difference of accuracy is within 0.01. Hence ensemble didn't achieve better performance.
I suspect the reason behind it is that the train data is already generalized enough that variance is not high. Hence using bagging may not be helpful to improve the accuracy

2 Part 2

2.1 Q1 - Formal Description

In the second part, we improve the IRT from one parameter to three parameters. Our aim is to improve the optimization. This is done by adding a discrimination parameter (g_j) and a guessing parameter (a_j). Discrimination parameter is utilized to assess the extent to which the items differentiate between various degrees or levels of the underlying trait and guessing parameter represents the probability of correctly answering the question randomly. So in our model, the discrimination parameter represent how a specific question can distinguish student ability. Therefore, questions with high discrimination are better at distinguishing students with strong ability and students with low ability. The guessing parameter represents the probability for a student to answer the question correctly by purely guesses. Since all questions are multiple choices questions with 4 choices, the guessing parameter is set to 0.25 for all questions by default.

$$
\begin{aligned}
L(C|\theta,\beta,g,\alpha) &= p(C|\theta,\beta,g,\alpha) \\
&= \prod_{i=1}^{d}\prod_{j=1}^{n}\left(g_j + (1-g_j)\frac{exp(a_j(\theta_i-\beta_j))}{1+exp(a_j(\theta_i-\beta_j))}\right)^{c^{ij}}\left(1-g_j-(1-g_j)\frac{exp(a_j(\theta_i-\beta_j))}{1+exp(a_j(\theta_i-\beta_j))}\right)^{1-c^{ij}} \\
&= \prod_{i=1}^{d}\prod_{j=1}^{n}\left(\frac{g_j+exp(a_j(\theta_i-\beta_j))}{1+exp(a_j(\theta_i-\beta_j))}\right)^{c_{ij}}\left(\frac{1-g_j}{1+exp(a_j(\theta_i-\beta_j))}\right)^{1-c_{ij}} \\
&= \prod_{i=1}^{d}\prod_{j=1}^{n}\frac{(g_j+exp(a_j(\theta_i-\beta_j)))^{c_{ij}}(1-g_j)^{1-c_{ij}}}{1+exp(a_j(\theta_i-\beta_j))}
\end{aligned}
$$

$$
\begin{aligned}
\ell(C|\theta,\beta,g,\alpha) &= log\left(\prod_{i=1}^{d}\prod_{j=1}^{n}\frac{(g_j+exp(a_j(\theta_i-\beta_j)))^{c_{ij}}(1-g_j)^{1-c_{ij}}}{1+exp(a_j(\theta_i-\beta_j))}\right) \\
&= \sum_{i=1}^{d}\sum_{j=1}^{n} c_{ij}log(g_j+exp(a_j(\theta_i-\beta_j))) + (1-c_{ij})log(1-g_j) - log(1+exp(a_j(\theta_i-\beta_j)))
\end{aligned}
$$

$$
\frac{\partial\ell(C|\theta,\beta,g,\alpha)}{\partial\theta_i} = \sum_{j=1}^{n}\left(c_{ij}a_j\frac{exp(a_j(\theta_i-\beta_j))}{g_j+exp(a_j(\theta_i-\beta_j))} - a_j\frac{exp(exp(a_j(\theta_i-\beta_j)))}{1+exp(exp(a_j(\theta_i-\beta_j)))}\right)
$$

$$
\frac{\partial\ell(C|\theta,\beta,g,\alpha)}{\partial\beta_j} = \sum_{i=1}^{d}\left(-c_{ij}a_j\frac{exp(a_j(\theta_i-\beta_j))}{g_j+exp(a_j(\theta_i-\beta_j))} + a_j\frac{exp(a_j(\theta_i-\beta_j))}{1+exp(a_j(\theta_i-\beta_j))}\right)
$$

$$
\frac{\partial\ell(C|\theta,\beta,g,\alpha)}{\partial\alpha_j} = \sum_{i=1}^{d}\left(c_{ij}(\theta_i-\beta_j)\frac{exp(a_j(\theta_i-\beta_j))}{g_j+exp(a_j(\theta_i-\beta_j))} - (\theta_i-\beta_j)\frac{exp(a_j(\theta_i-\beta_j))}{1+exp(a_j(\theta_i-\beta_j))}\right)
$$

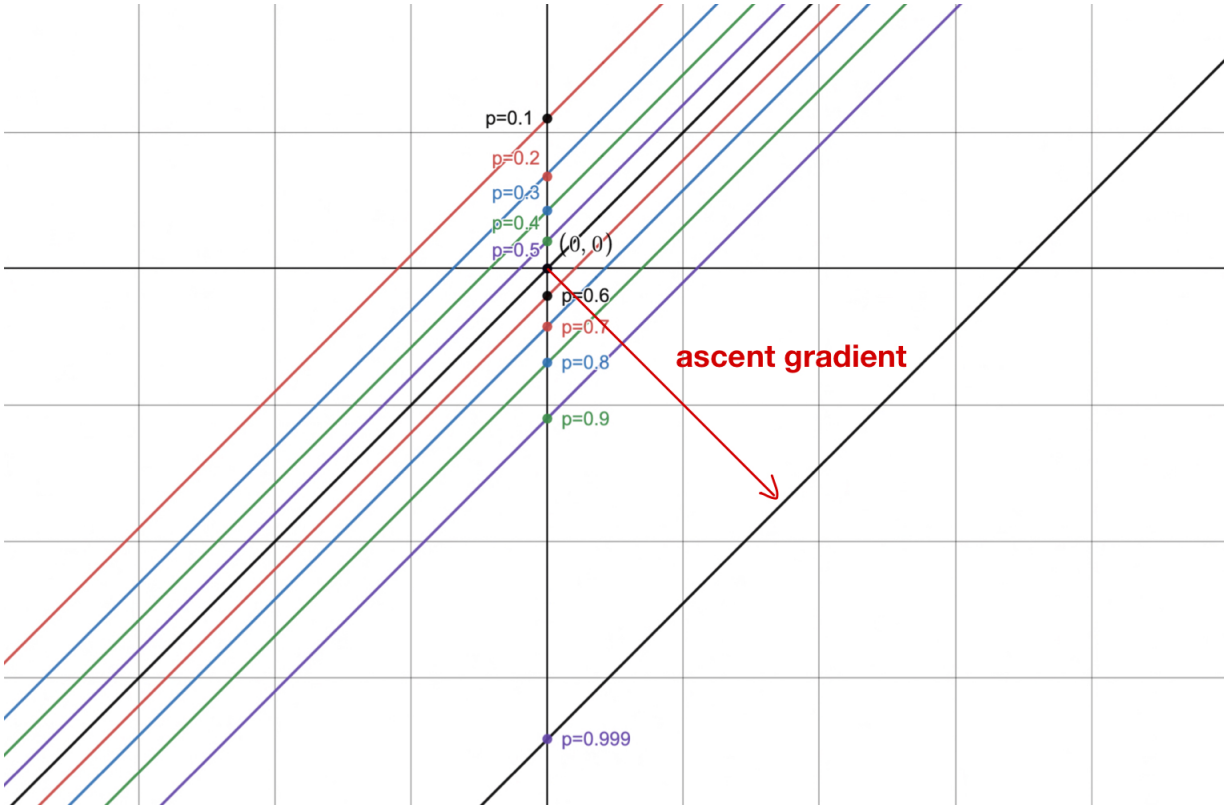
$$
\frac{\partial\ell(C|\theta,\beta,g,\alpha)}{\partial_j} = \sum_{i=1}^{d}\left(\frac{c_{ij}}{g_j+exp(a_j(\theta_i-\beta_j))} - \frac{1-c_{ij}}{1-g_j}\right)
$$

2.2 Q2 - Figure or Diagram

In part(a), we aim to maximize the likelihood:

$$\prod_{i,j} p(c_{i,j}|\theta_i, \beta_j) = \prod_{i=1}^{d}\prod_{j=1}^{n}\left(\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}\right)^{c^{ij}}\left(1 - \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}\right)^{1-c^{ij}}$$

with an ascent gradient algorithm. From the likelihood model, we can see that the probability result depends on 2 parameters: θ_i and β_j. We initialize those 2 parameters at 0, and then change them a bit in each iteration of ascent gradient algorithm until it reaches their optimal (or near-optimal) values at which the likelihood is maximized. The contour of our part(a) model can be shown below:



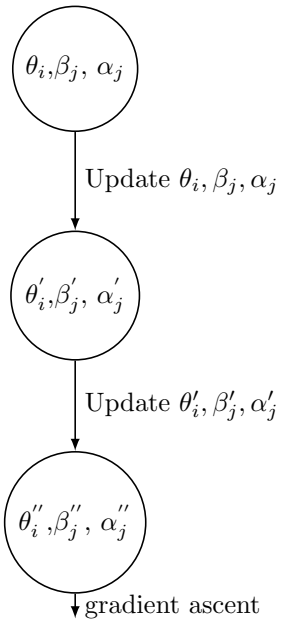
Because the model is multivariate with $(d + n)$ variables, it is hard to visualize the actual contour. To simplify but without any loss, we can just use straight-lines to represent them. The arrow in the above graph represents the direction that our parameters θ and β march during the ascent gradient algorithm.

In part(b), we extends this model by including 2 more parameters: guess and α, where $guess_j$ represents how probable a problem j can be guessed correctly, and α_j represents a discrimination bias of question j. The model is shown below:

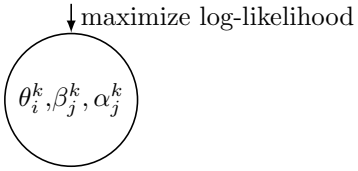
$$\prod_{i,j} p(c_{i,j}|\theta_i, \beta_j, guess_j, \alpha)j) = \prod_{i=1}^{d}\prod_{j=1}^{n}\left(g_j + (1 - g_j)\frac{exp(a_j(\theta_i - \beta_j))}{1 + exp(a_j(\theta_i - \beta_j))}\right)^{c^{ij}}\left(1 - g_j - (1 - g_j)\frac{exp(a_j(\theta_i - \beta_j))}{1 + exp(a_j(\theta_i - \beta_j))}\right)^{1-c^{ij}}$$

With the same idea, we initialize θ_i and β_j, α_j to be 0, $guess_j$ to be 0.25, and $alpha_j$ to be 1. Next, we update three of those parameters θ_i, β_j, and α_j using the same ascent gradient algorithm as above to maximize the likelihood.

Furthermore, we can use a flow chart to visualize it in another way: $\forall i, j \in \mathbb{N}$ by $1 \leq i \leq d, 1 \leq j \leq n$



...after k iterations...



2.3 Q3 - Comparison or Demonstration

Table: Accuracy Comparison Across Models

Model	**Validation Accuracy**	**Test Accuracy**
KNN	0.6922	0.6816
IRT	0.7070	.7053
AutoEncoder	0.6874	0.6892
Extended IRT	0.7083	0.7053
Ensemble	0.6999	0.7002

As shown from above table, the extended IRT model generates the highest accuracy among all choices.

Experiment:
We hypothesize that accuracy of our model will be increased due to optimization by adding 2 more parameters.
Therefore, we have designed another model with only adding one discrimination parameter a_i with probability that the question j is correctly answered by student i is formulated as:$p(c_{ij} = 1|\alpha_i, \theta_i, \beta_j) = \frac{exp(\alpha_i * \theta_i - \beta_j)}{1+exp(\alpha_i * \theta_i - \beta_j)}$ based on formula of "2-parameter" IRT model in https://hummedia.manchester.ac.uk/institutes/methods-manchester/docs/irt.pdf. And we will compare the result of two models.

Table: Accuracy Comparison Between two Models

Model	**Validation Accuracy**	**Test Accuracy**
IRT with α	0.70745	0.70505
IRT with α, g	0.70829	0.70533

From the results above, we can find that the fully extended model is slightly better than "2-parameter" IRT model. Hence we can see that adding guessing parameter can slightly improve the performance.

2.4 Q4 - Limitation

We can see that even if we added two parameters to IRT models, the accuracy only increase very small amount. Hence there are still some limitations of our extension:

1. We didn't make use of subject.meta and question.meta. Perhaps better model can be obtained because students ability to choose correct answer can vary based on their advantage on certain fields.

2. Guessing parameter for each question was fixed on 0.25 becasue we assume for each question the possibility of guessing the answer correctly is 0.25. However, for some questions, depending on the design, even if students has no knowledge about the subject, they can still use exclusive method to increase the correctness, which means the guessing parameter should be increased.

3. We didn't perform regularization on each gradient ascent. It is always possible that doing regularization will improve the performance of the model.

4. This model will not doing well with limited data. Because students ability can not be determined correctly with very few number of questions.

Possible extensions: We can add parameter corresponding to students ability on each subject. We can add regularization to each parameter. And we can update guessing parameter in a more approprite way.

3 References

- Nathan Thompson, P. D. (2022, December 21). What is the three parameter IRT model (3PL)? Assessment Systems. Retrieved March 31, 2023, from https://assess.com/three-parameter-irt-3pl-model/
- Shryane, N. (n.d.). What is item response theory? - university of manchester. What is Item Response Theory? Retrieved March 31, 2023, from https://hummedia.manchester.ac.uk/institutes/methods-manchester/docs/irt.pdf
- Wood, J. (2017, November 12). Logistic IRT models. Retrieved March 31, 2023, from https://quantdev.ssri.psu.edu/sites/qdev/files/IRT_tutorial_FA17_2.html