In this part, We modified one-parameter IRT implemented. Since from IRT, we could see there is no big difference between validation accuracy and test accuracy, indicating that we are not overfitting. So here we are trying to improve the model by adding more features to our model to increase the test accuracy. Our algorithm extension include two parts:

(1) adding discrimination parameter α_j for each question j
(2) initialize θ by gender group.

0.1 Formal Description

(1) Adding discrimination parameter α_j
Problem in Part A:
In Part A, we have used one-parameter IRT model with β_j representing the difficulty of the question j, and θ_i representing the i-th students ability. β_j is determined at the point of median point, and the question discrimination, i.e., the rate at which the probability of endorsing a correct question changes given ability levels, is fixed for all questions. However, with same discrimination for all questions, we cannot detect subtle differences in the ability of the respondents, which is not ideal. Since for some questions, they have a higher discrimination, and students with a higher ability can do much better than students with a lower ability. While for other questions, the discrimination is small, and students with different ability will perform similarly on these questions. For example, in figure 1, we have fitted the 1-parameter model using Part A Q2, and plot $p(c_i = 1|\theta_i) = \frac{exp(\theta_i - \beta)}{1+exp(\theta_i - \beta)}$ using β for each question. And each point in the figure represents the probability that the question is correctly answered given the student's ability using the data in train_data.csv. From the figure, it is clear that the discrimination for question 1 is small, since when students' ability is small(-1), the probability that it can be answered correctly is around 0.8. However, even when the ability increases, the probability does not increase and it even decreases. While for Question 45, the discrimination would be quite big, since from the plot, it is very clear that probability of answering the question correctly is almost 1 for ability greater than 0, and almost 0 when ability is lower than -1. But since we are using 1-parameter model, the discrimination is the same for all questions, so from Figure 1, curves for these 3 different questions have the same shape, and the only difference is that the position of the median point is not different, and it does not fit the data very well.

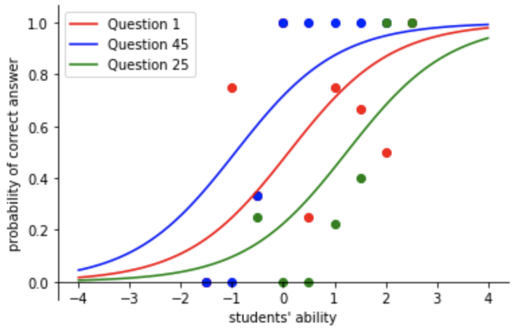


Figure 1: 1-Parameter Model

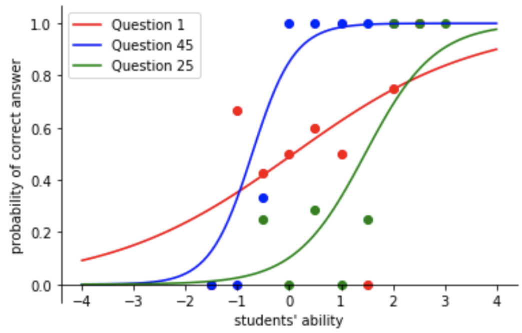


Figure 2: 2-Parameter Model

Extension:
So we decided to introduce discrimination α_i for each question i in our new model. So different questions will have different slopes, and the steeper the slope, the higher the discrimination, and this question can detect subtle differences in respondents' ability. And our new model uses $p(c_i = 1|\theta_i) = \frac{exp(\alpha_i(\theta_i - \beta))}{1+exp(\alpha_i(\theta_i - \beta))}$. And this new model can fit the data better. From Figure 2, using 2-parameter model, the curve for Question 1 is relatively flat, since as we have argued above that Question 1 cannot discriminate the probability of answering this question based on students' ability very well. And slope for Question 45 is very large, since Question 45 can greatly detect difference in probability based on students' ability. In comparison to Figure 1, the model in Figure 2 fits the data much better.
The way we implemented the 2-parameter IRT model is similar to what we have done in Part A, i.e., performing alternating gradient descent on θ, β, α to maximize the log-likelihood. And the comparison between 1-parameter model and 2-parameter model is listed below.

	1-parameter IRT	2-parameter IRT
probability that question j is correctly answered	$p(c_{ij} = 1 \mid \theta_i, \beta_j) = \frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)}$	$p(c_{ij} = 1 \mid \theta_i, \beta_j, \alpha_j) = \frac{exp[\alpha_j(\theta_i - \beta_j)]}{1+exp[\alpha_j(\theta_i - \beta_j)]}$
log-likelihood for all students and questions	$\log(\mathbf{C} \mid \theta, \beta) = \sum_{i=1}^{542} \sum_{j=1}^{1774} [c_{ij} log(\frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)}) + (1 - c_{ij}) log(1 - \frac{exp(\theta_i - \beta_j)}{1+exp(\theta_i - \beta_j)})]$	$\log(\mathbf{C} \mid \theta, \beta, \alpha) = \sum_{i=1}^{542} \sum_{j=1}^{1774} [c_{ij} log(\frac{exp(\alpha_j\theta_i - \alpha_j\beta_j)}{1+exp(\alpha_j\theta_i - \alpha_j\beta_j)}) + (1 - c_{ij}) log(1 - \frac{exp(\alpha_j\theta_i - \alpha_j\beta_j)}{1+exp(\alpha_j\theta_i - \alpha_j\beta_j)})]$
derivative with respect to θ_i	$\sum_{j=1}^{1774}(c_{ij} - \frac{e^{\theta_i - \beta_j}}{1+e^{\theta_i - \beta_j}})$	$\sum_{j=1}^{1774}(\alpha_j c_{ij} - \alpha_j \frac{e^{\alpha_j\theta_i - \alpha_j\beta_j}}{1+e^{\alpha_j\theta_i - \alpha_j\beta_j}})$
derivative with respect to β_j	$\sum_{i=1}^{542}(-c_{ij} + \frac{e^{\theta_i - \beta_j}}{1+e^{\theta_i - \beta_j}})$	$\sum_{i=1}^{542}(-\alpha_j c_{ij} + \alpha_j \frac{e^{\alpha_j\theta_i - \alpha_j\beta_j}}{1+e^{\alpha_j\theta_i - \alpha_j\beta_j}})$
derivative with respect to α_j		$\sum_{i=1}^{542}(c_{ij}\theta_i - c_{ij}\beta_j - (\theta_i - \beta_j)\frac{e^{\alpha_j\theta_i - \alpha_j\beta_j}}{1+e^{\alpha_j\theta_i - \alpha_j\beta_j}})$

(2) Initialize theta by gender group

Problem in Part A: In part A, we initialize every student ability(θ) to zero which we are treating every student's ability equally at first. Our intuition here is that different gender of student may lead to different ability, like some people are tend to believe boys may perform better than girls on STEM subjects. So here we are trying to initialize the θ according to student's gender. The basic process is to first separate the training data into 3 sets according to gender, and train each set to gain the approximation of that gender's θ. Finally training the model using the whole training data and initialize the theta using the approximation we calculated for each gender.

Algorithm box:

Step 1: improve the irt model by adding a new parameter.

```
def update_theta_beta_alpha(data, lr, θ, β, α):
    θ_i = θ_i − lr* gradient of θ_i
    β_j = β_j − lr* gradient of β_j
    α_j = α_j − lr* gradient of α_j
return θ, β, α
```

Step 2: split the training data according to the user's gender

```
def split_data_gender(training_data):
    # we are going to split data according to their gender
    for data in training_data:
        gender = find_gender(data["user_id"])
        if gender is 0:
            put data in gender0
        else if gender is 1:
            put data in gender1
        else:
            put data in gender2
    return gender0, gender1, gender2
```

Step 3: train each gender data using the improved irt model and get the corresponding theta for each gender group.

```
def theta_initializer(gender0, gender1, gender2):
    # this list stores the respective initializer for each gender group
    theta_initializer = []
    # we are going to split data according to their gender
    for i in 0,1,2:
        theta_list = improved_irt(gender_i)
        add average value of theta_list to theta_initializer
    return theta_initializer
```

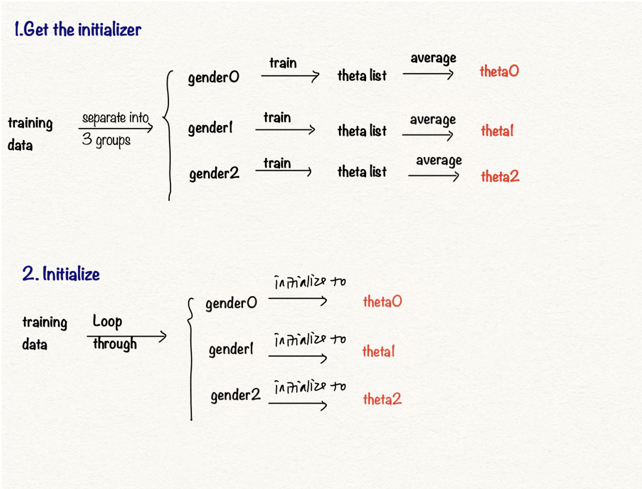
Step 4: This is our **final model**, which created based on imporved irt model while here each theta is initialized according to their gender.

```
def final_irt(train_data, theta_initializer):
    # initialize the theta according to gender
    for data in train_data:
        # identify gender of the user
        gender = find_gender(data["user_id"])
        if gender is 0:
```

```
            initialize it to theta_initializer[0]
        else if gender is 1:
            initialize it to theta_initializer[1]
        else:
            initialize it to theta_initializer[2]
    #other things remain the same as the improved_irt
```

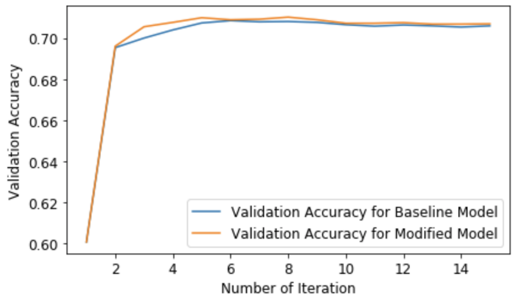
0.2 Diagram

Simple model for initialize theta by gender group.



Please refer to partb.py and the algorithm box above for the detailed implementation.

0.3 Comparison



	Baseline Model	Modified Model
Final Test Accuracy	0.7058989556872707	0.7106971493084956
Final Test Accuracy on Gender0(Unspecified)	0.6836734693877551	0.6938775510204082
Final Test Accuracy on Gender1(Female)	0.7073007367716009	0.7079705291359679
Final Test Accuracy on Gender2(Male)	0.7155425219941349	0.7221407624633431

The test accuracy using the modified model is 0.7106971493084956, and there is a 0.4% improvement on overall test accuracy using the modified model.
The overall purpose of using the modified model is to
1. Improve overall performance by using different discrimination for different questions.
2. Improve performance for unspecified, female, male group by using different initializations for each group.
From the table above, it is obvious that the overall test accuracy has improved.
And to further see whether accuracy on each group has improved, we further calculate the test accuracy for each group separately. Since test accuracy on each group has all increased, our model has indeed improved performance on overall model, as well as on each group, which is consistent with our hypothesis.

0.4 Limitations and Extension

- In 2-parameter IRT, we have not taken group variables such as gender, question subject into account, so if the α, β are quite different for each group, our model would not perform well. Although we have tried

to separate genders into three groups: 0(unspecified), 1(female), 2(male), and fit three different β, α for each group, it does not work well using the given data.
One possible reasons is that the training data is not large enough, since if we separate training data into three groups, there are some cases where no student in a group have answered a specific question, thus the α, β for this question and group will remain unchanged from initialization.
A possible extension is to use more data to train the model or to perform a leave-one-out cross validation to solve the problem of lacking data, and then to fit different α, β, θ for each group.

- We didn't make use of the question_meta to improve our algorithm. So our algorithm may approach poorly when a student ability is quite unbalanced. For example a student is very good at solving algebra questions but are very bad at solving problems related to graph. If majority training data we get for this student is about the algebra, we might think the student a high theta. So when the test data is about graph, we may probably make a prediction that it would answer the question correctly. However in this case we would probably make a mistake.
  A possible extension would be make several groups of questions according the subjects they are belong to. Then get a approximation of parameters β and α for each group and make a different initialization during the training process.