CSC311 Final Project

Hanchun Wang(1005399720)
Yi Chen (1003849801)

May 1, 2021

Problem description

Online education services, such as Khan Academy and Coursera, provide a broader audience with access to high-quality education. On these platforms, students can learn new materials by watching a lecture, reading course material, and talking to instructors in a forum. However, one disadvantage of the online platform is that it is challenging to measure students' understanding of the course material. To deal with this issue, many online education platforms include an assessment component to ensure that students understand the core topics. The assessment component is often composed of diagnostic questions, each a multiple choice question with one correct answer. The diagnostic question is designed so that each of the incorrect answers highlights a common misconception. An example of the diagnostic problem is shown in figure 1. When students incorrectly answer the diagnostic question, it reveals the nature of their misconception and, by understanding these misconceptions, the platform can offer additional guidance to help resolve them.

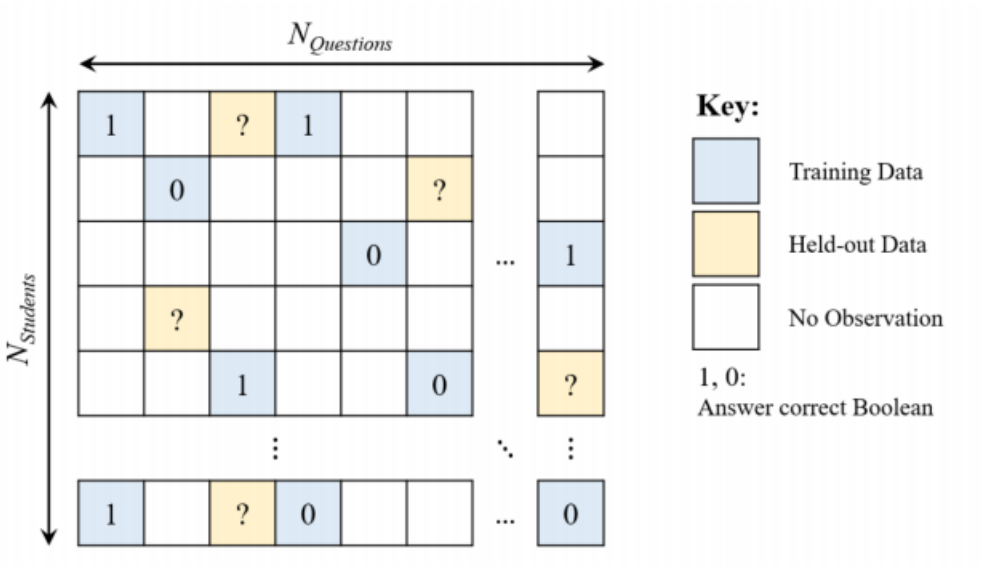


Figure 2: An example sparse matrix [1].

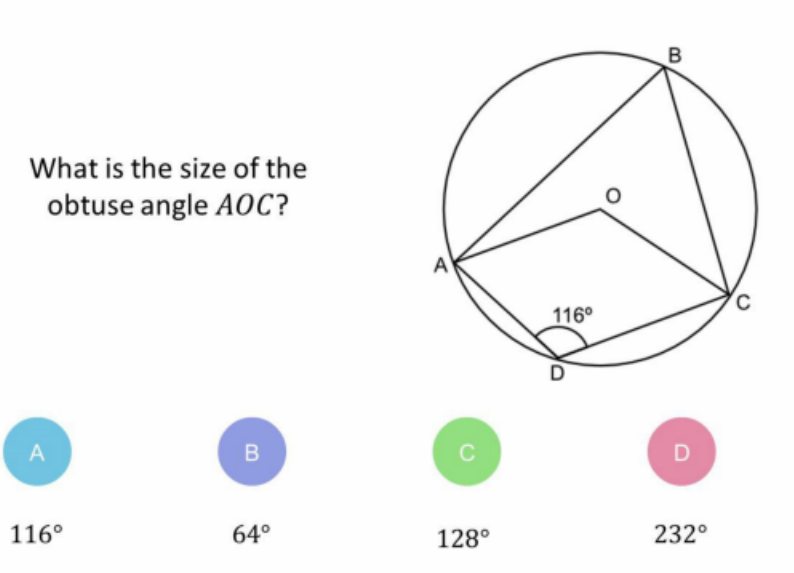


Figure 1: An example diagnostic question [1].

In this project, you will build machine learning algorithms to predict whether a student can correctly answer a specific diagnostic question based on the student's previous answers to other questions and other students' responses. Predicting the correctness of students' answers to as yet unseen diagnostic questions helps estimate the student's ability level in a personalized education platform. Moreover, these predictions form the groundwork for many advanced customized tasks. For instance, using the predicted correctness, the online platform can automatically recommend a set of diagnostic questions of appropriate difficulty that fit the student's background and learning status.

You will begin by applying existing machine learning algorithms you learned in this course. You will then compare the performances of different algorithms and analyze their advantages and disadvantages. Next, you will modify existing algorithms to predict students' answers with higher accuracy. Lastly, you will experiment with your modification and write up a short report with the results.

You will measure the performance of the learning system in terms of prediction accuracy, although you are welcome to include other metrics in your report if you believe they provide additional insight:

$$\text{Prediction Accuracy} = \frac{\text{The number of correct predictions}}{\text{The number of total predictions}}$$

Solution algorithm and analysis

Intuition

We choose item response theory to be our base model since we believe it closely address the problem. Due to our reality experience, we believe questions may also have difficulty in each subject and students may have abilities in each subject. Therefore we can first apply item response theory on each subject. Then for each student i and question j, we get a bunch of ability difference for student i solving question j. We use these numbers to determine final prediction that whether student i can correctly answer question j. This is a binary classification problem so here we can use logistic regression.

To improve the performance, we also try the following:

1. Multiply some constant to ability difference calculate by each subject.

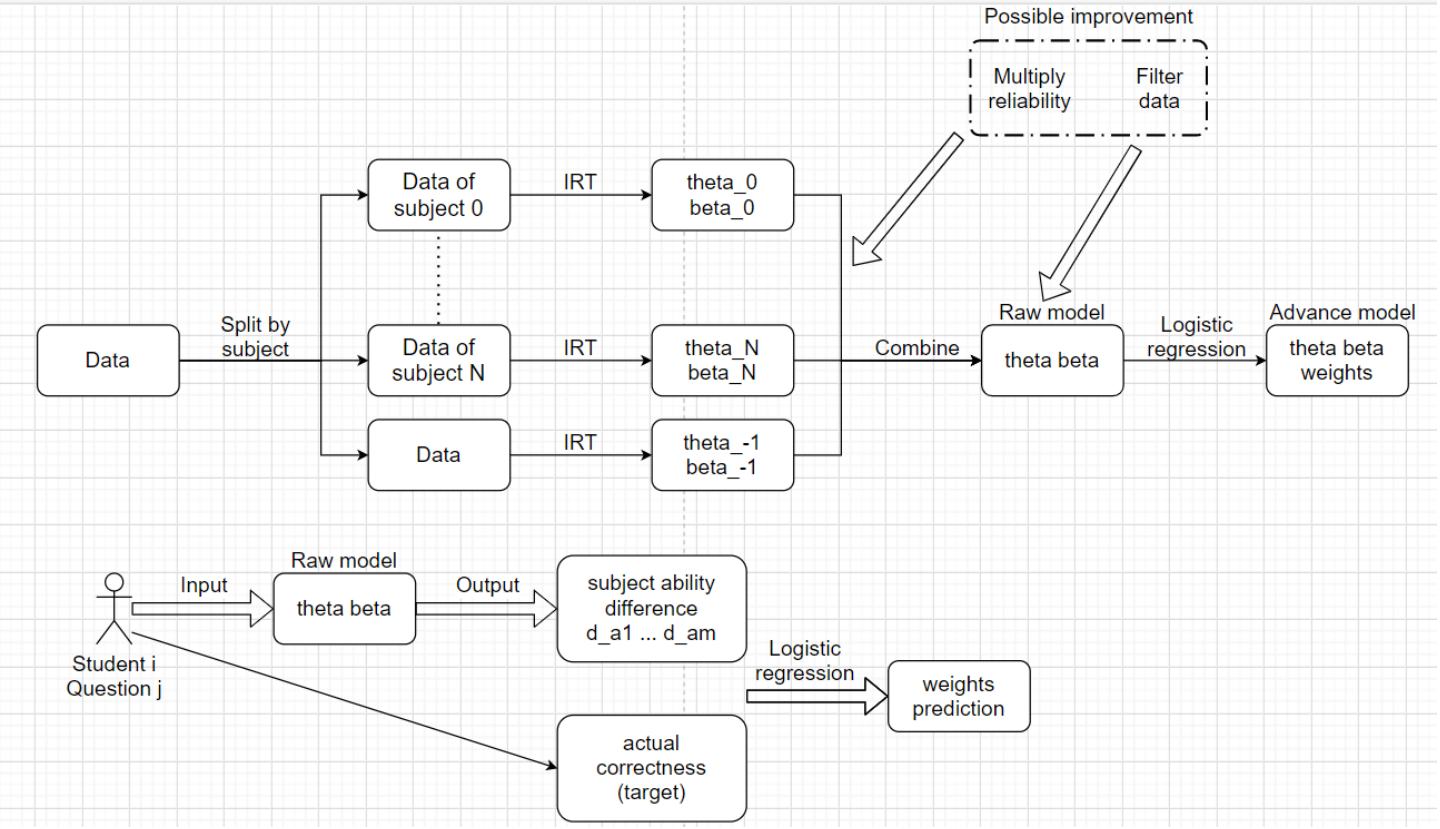
   This is because we use fewer data to train a IRT model for each subject. So perhaps the model of each subject is less reliable.

2. Filter out some data points which are likely to be "guessing answer"

   Student may guess the answer when the question is too hard for them, and they may accidentally guess the correct answer. But these data are spam and will mislead the model, so we try to filter out these kind of data points.

   We done this by pop out data where all subject ability difference is less than some threshold(for example 0), but the answer is correct.

Idea diagram



Expectation

1. We believe our base model may be under-fitting. Since our final goal have more expressive hypothesis space, it should prevent under-fitting. Therefore we believe it will improve optimization.
2. Because we use many subjects ability difference. We think this result should be more stable and variance is reduced.

Algorithm

How model is trained:

`train_Advanced_IRT`($data, num\_subject$)
$theta, beta \leftarrow$ empty matrix
$theta[:, num\_subject], beta[:, num\_subject] \leftarrow$ Apply IRT on data of subject i
for $i = 0, ..., num\_subject - 1$ **do**
  $theta[:, i], beta[:, i] \leftarrow$ Apply IRT on data of subject i
end for
$weights \leftarrow$ empty dictionary
for question q in $data$ **do**
  $weights[q] \leftarrow$ `Logistic_Regression`(data of question i, $q, theta, beta$)
end for
return $weights, theta, beta$

`Logistic_Regression`($data, q, theta, beta$)
$matrix \leftarrow$ empty matrix
$label \leftarrow$ empty array
for data point s, c in data **do**
  $lst \leftarrow$ empty list
  for subject i of question q **do**
    lst append $theta[s, i] - beta[q, i]$
  end for
  $matrix$ add a new row lst
  $label$ add a new entry c
end for
return use logistic regression with L2 regularization to calculate weights base on input $matrix$ and $label$

How we compute probability of each data point using trained model:

`predict_probability`($weights, theta, beta, q, s$)
$d \leftarrow 0$
for subject i of question q **do**
  $d \leftarrow d + weights[q][i] * (theta[s, i] - beta[q, i])$
end for
return `sigmoid`(d)

Improvement filter:

We pop out every training data point which is spam.

`is_spam`($theta, beta, threshold, q, s, correctness$)
if correctness $= 1$ **then**
  for subject i of question q **do**
    if $theta[s, i] - beta[q, i] > threshold$ **then**
      return FALSE
    end if
  end for
  return TRUE
end if
return FALSE

Improvement reliability:

Whenever we try to compute quantity $theta[s, i] - beta[q, i]$, we multiply this quantity by a reliability hyper-parameter.

Base model and result

learning rate: 0.01, number of iterations: 20.

```
Validation accuracy: 0.7064634490544736
Test accuracy: 0.7050522156364663
```

Final model and result

IRT learning rate: 0.01, number of iterations: 20.

Reliability constant for each subject: 0.01, filter threshold: -0.1.

Logistic regression learning rate: 1, number of iterations: 50, regularization coefficient: 1.

```
41735 55299
Training accuracy: 0.7547152751405993
5009 7086
Validation accuracy: 0.7068868190798758
2495 3543
Test accuracy: 0.7042054755856618
```

Variance comparison

We generate a bunch of samples that always contain 2 particular data points i, j. We focus on prediction probability on student s_i answer question q_j. For each sample we use it to train one advance IRT model and one simple IRT model. We compare the variance of list of predictions for advanced IRT and simple IRT.

```
num_model: 10 sample size: 1000
AdvanceIRT Variance: 0.00022385718000316281
SimpleIRT Variance: 0.0007650007062995029
```

Conclusion made

1. The optimization is rarely improved and for most time is decreased. So our hypothesis failed.

   Maybe the all other subjects ability is less important than overall ability. Notice the fact that all question samples are of subject 0, which are math problems. So math ability maybe always is the most important factor so it is not essential to consider other minor subjects.

   Another possible reason is that data grouped by subjects has far less data points than total training data, so subject ability may not be reliable. Even though we tried to multiply reliability constant, but we cannot guarantee it address the problem.

2. The variance is indeed become smaller. Our expectation is satisfied.

Limitations

1. Filter process is hard to filter "guessing answer" accurately.

   During our filter process, the data marked "guessing answer" is when student has less ability than question difficulty for all subjects. However correct "guessing answer" will make student ability become higher when we are doing IRT, Then those data may not be filtered.

   Also student can sometimes solve question that is only a little hard. But our filter process may filter this piece of data out and this student will result in lower ability which is not the reality.

2. Reliability factor is hard to determine.

   It is hard to tell how many data is enough to estimate a good model. If 100 data points is enough then we can fully trust both model with 100 data points and model with 10000 data points. But if instead 1000 data points is enough to estimate a good model, then we do not know how much we should trust the model with 100 data points.

3. IRT model is not enough expressive.

   Even if latent features ability and difficulty exist and can be precisely computed. The result probability curve may have significant difference from any IRT model curve.

Possible solution

1. To improve filter process

   On one hand, we may use another model to figure out which data points should be filtered out. On the other hand, we can try to get more data to address this problem. We can design questionnaire to ask students whether they frequently guess answers or not. We can even add "I don't know" option for each question to reduce "guessing answer". By doing this, we can obtain a priori distribution of students behavior in answering question, therefore, we can filter the noisy data bu applying a weighted L1 regularization.

2. To precisely determine reliability factor

   The most direct way is just to gather enough data. Then we do not even need to worried about reliability for any segment of data we used.

3. Further improve expressiveness of IRT model

   First, 3-parameter IRT may be helpful. More generally, we can instead observe how the probability curve looks like for given student solving a bunch of questions with different difficulty. Then we may give a generative model that is more closely address this problem.