CSC311 - Final Assignment

Zelong Liu, Fizzah Mansoor, Harrison Deng

December 3, 2021

Contents

Part I
Predicting Student Correctness

1 K-Nearest Neighbor

a Complete Main kNN, Plot and Report Accuracy

b Selecting k*

c Implementing Impute by Item

d Comparing user and item based Collaborative Filtering

e Potential Limitations of kNN in this Context

2 Item Response Theory

a Mathematical Derivations for IRT

We are given that $p(c_{ij} = 1|\boldsymbol{\theta}, \boldsymbol{\beta})$. We will assume c_{ij} is a value in $\boldsymbol{C}$ where i and j as coordinates are in set O as defined:

$$O = \{(i, j) : \text{Entry } (i, j) \text{ of matrix } \boldsymbol{C} \text{ and is observed}\}$$

.

Since this c_{ij} is a binary value, we can describe $P(\boldsymbol{C}|\boldsymbol{\theta}, \boldsymbol{\beta})$ with a bernoulli distribution:

$$p(C|\boldsymbol{\theta}, \boldsymbol{\beta}) = \prod_{ij} [\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}]^{c_{ij}} [\frac{1}{1 + exp(\theta_i - \beta_j)}]^{(1-c_{ij})}$$

Therefore, our Likelihood function is:

$$L(\boldsymbol{\theta}, \boldsymbol{\beta}) = \prod_{ij} [\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}]^{c_{ij}} [\frac{1}{1 + exp(\theta_i - \beta_j)}]^{(1-c_{ij})}$$

Then, apply log to obtain the log-likelihood where N and M are the number of users and questions respectively:

$$\begin{aligned}
L(\boldsymbol{\theta}, \boldsymbol{\beta}) &= \prod_{ij} [\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}]^{c_{ij}} [\frac{1}{1 + exp(\theta_i - \beta_j)}]^{(1-c_{ij})} \\
log(L(\boldsymbol{\theta}, \boldsymbol{\beta})) &= \log(\prod_{ij} [\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}^{c_{ij}}] [\frac{1}{1 + exp(\theta_i - \beta_j)}^{1-c_{ij}}] \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} \log([\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}^{c_{ij}}] [\frac{1}{1 + exp(\theta_i - \beta_j)}^{1-c_{ij}}]) \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij}((log(exp(\theta_i - \beta_j)) - log(1 + exp(\theta_i - \beta_j))) \\
&+ (1 - c_{ij})(\log(1) - \log(1 + exp(\theta_i - \beta_j))) \\
&= \sum_{i=1}^{N} \sum_{j=1}^{M} [c_{ij}(\theta_i - \beta_j) - \log(\frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)})]
\end{aligned}$$

Then, we solve for the partial derivative with respect to θ_i and β_j respectively:

$$\frac{\delta}{\delta \theta_i} = \sum_{j=1}^{M} [c_{ij} - \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}]$$

$$\frac{\delta}{\delta \beta_j} = \sum_{i=1}^{N} [-c_{ij} + \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}]$$

b Implementation of IRT

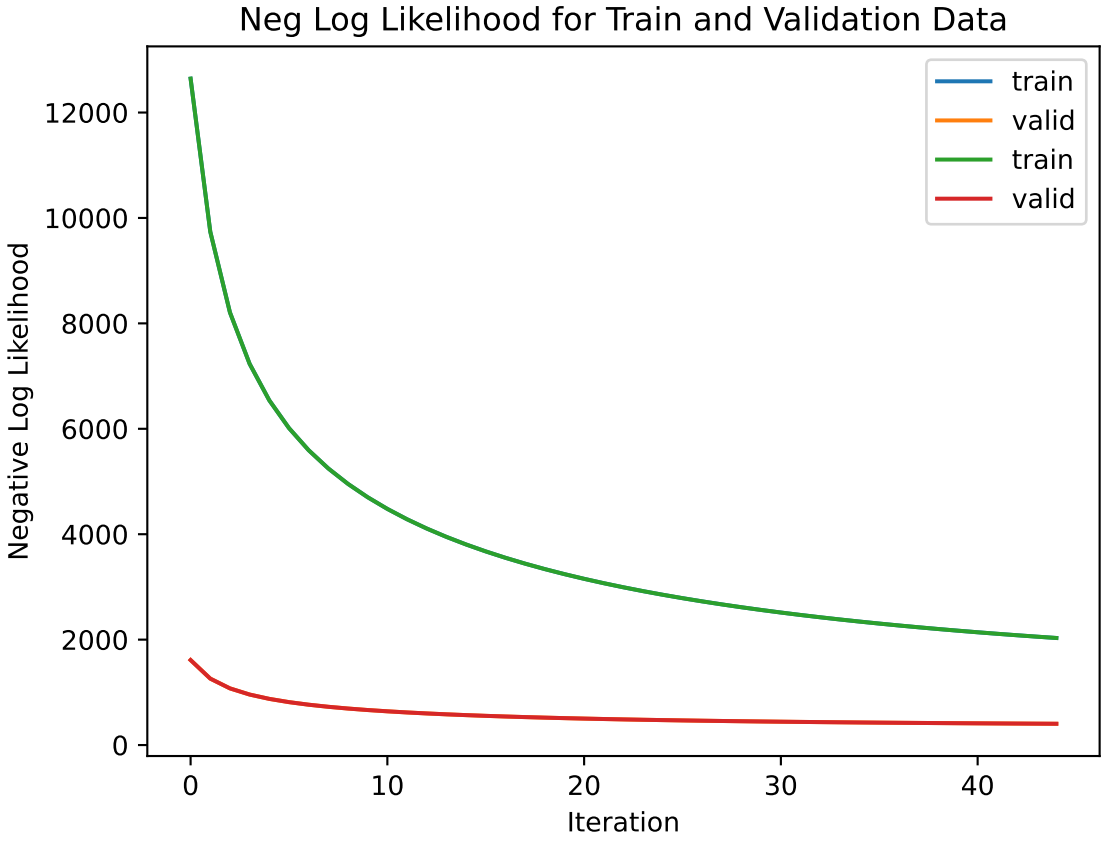
The implementation of IRT is in `part_a/item_response.py`. We chose the hyperparameters α and iterations number by performing multiple combinations of them and seeing which one had the highest validation score (automated, see mentioned code file for this automation). We then manually adjusted the set of tested values and repeated. Doing this a few times resulted in:

```
import codebase.part_a.item_response as irt
print()
irt_results = irt.main("./codebase/data")


Training with lr of 0.005 and 45 number of iterations.
Final accuracy: 0.7056167090036692
lr*: 0.005 iterations*: 45 acc*: 0.7056167090036692
test accuracy: 0.7042054755856618
```

The following is a plot of the training curve (and the code used to generate it):

```
plot(irt_results["train_nllks"], label="train")
plot(irt_results["val_nllks"], label="valid")
ylabel("Negative Log Likelihood")
xlabel("Iteration")
title("Neg Log Likelihood for Train and Validation Data")
legend()
```



c Reporting Validation and Test Accuracies

Validation and test accuracies have been calculated in the previous call to the main function. Again, implementation is in `part_a/item_response.py`.

Our validation accuracy:

```
print(irt_results["val_acc"])
```

```
0.7056167090036692
```

Test accuracy:

```
print(irt_results["test_acc"])
```

```
0.7042054755856618
```

d Plots of Questions With Respect to θ and β

3 Neural Networks

a Differences Between ALS and Neural Networks

b Implementing AutoEncoder

c Tuning and Training NN

d Ploting and Reporting

e Implementing L_2 Regularization

4 Ensemble

Part II
Modifying for Higher Accuracy

5 Formal Description

6 Figure or Diagram

7 Comparison or Demonstration