CSC311 - Final Assignment

Zelong Liu, Fizzah Mansoor, Harrison Deng

December 2, 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:

$$\begin{aligned}
\frac{\delta}{\delta\theta_i} &= \sum_{j=1}^{M}[c_{ij} - \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}] \\
\frac{\delta}{\delta\beta_j} &= \sum_{i=1}^{N}[-c_{ij} + \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}]
\end{aligned}$$

b Implementation of IRT

The implementation of IRT is in `part_a/item_response.py`. We chose the hyperparameters α and iterations via the code below:

```
#TODO: Import
iteration_sets = [45, 180, 185, 190]
lrs = [0.001, 0.005]

lr_star = None
iterations_star = None
acc_star = 0
for iterations in iteration_sets:
    for lr in lrs:
        print("Training with lr of {} and {} number of iterations.".format(lr, iterations))
        theta, beta, step_accs = irt(sparse_matrix, val_data, lr, iterations, verbosity=2)
        acc = evaluate(val_data, theta, beta)
        print("Final accuracy: {}".format(acc))
        if acc > acc_star:
            acc_star = acc
            lr_star = lr
            iterations_star = iterations
print("lr*: {} iterations*: {} acc*: {}".format(lr_star, iterations_star, acc_star))
```

c Reporting Validation and Test Accuracies

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