CSC311 Project Final Report

Group Members:
Zixuan Zeng 1008533419

August 1, 2024

Part A

Q1 - KNN

1.(a)

```
knn_impute_by_user:
Validation Accuracy: 0.6244707874682472
Validation Accuracy: 0.6780976573525261
Validation Accuracy: 0.6895286480383855
Validation Accuracy: 0.6755574372001129
Validation Accuracy: 0.6692068868190799
Validation Accuracy: 0.6522720858029918
```

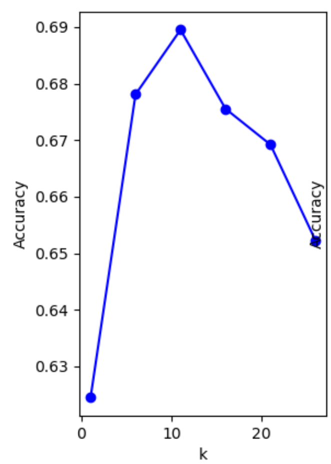

Figure 1: Accuracy vs k for KNN Impute by User

1.(b)

```
Chosen argmax k*: 11 , Test accuracy: 0.6841659610499576
```

Figure 2: Test accuracy with k*

1.(c)

The underlying assumption is that if question A is answered similarly by many students as question B, A's predicted response from specific students matches that of question B.

```
knn_impute_by_item:
Validation Accuracy: 0.607112616426757
Validation Accuracy: 0.6542478125882021
Validation Accuracy: 0.6826136042901496
Validation Accuracy: 0.6860005644933672
Validation Accuracy: 0.6922099915325995
Validation Accuracy: 0.69037538808919
```

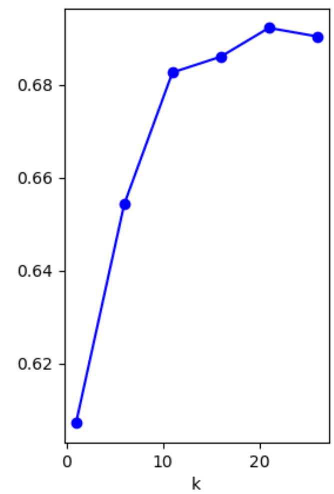

Figure 3: Accuracy vs k for KNN Impute by Item

```
Chosen argmax k*: 21 , Test accuracy: 0.6683601467682755
```

Figure 4: Test Accuracy with k*

1.(d)

The test accuracy for the user-based method (0.6842) is higher than that for the item-based method (0.6684). Therefore, the user-based collaborative filtering method performs better than the item-based collaborative filtering method in this case.

1.(e)

- Computationally expensive. KNN practically has no training process. With large datasets, as the number of students/questions grow, the time required to compute the distances and to identify the nearest neighbors at test time grows significantly.
- Curse of Dimensionality. When the sparse_matrix has too many missing values, it's hard to find good nearest neighbors, since most points will be about the same distances. This affects the prediction accuracy.

Q2 - IRT

2.(a)

$$p(c_{ij} = 1|\theta_i, \beta_j) = \frac{\exp(\theta_i - \beta_j)}{1 + \exp(\theta_i - \beta_j)} = \sigma(\theta_i - \beta_j)$$

log-likelihood:

$$\log p(C|\theta, \beta) = \sum_i \sum_j [c_{ij}(\theta_i - \beta_j) - \log(1 + \exp(\theta_i - \beta_j))]$$

The derivative of the log-likelihood with respect to θ_i:

$$\frac{\partial \log p(C|\theta, \beta)}{\partial \theta_i} = \sum_j [c_{ij} - \sigma(\theta_i - \beta_j)]$$

The derivative of the log-likelihood with respect to β_j:

$$\frac{\partial \log p(C|\theta, \beta)}{\partial \beta_j} = \sum_i [-c_{ij} + \sigma(\theta_i - \beta_j)]$$

2.(b)

```
# hyper-parameters
lr = 0.008
iterations = 100
```

Figure 5: Hyper-Parameters

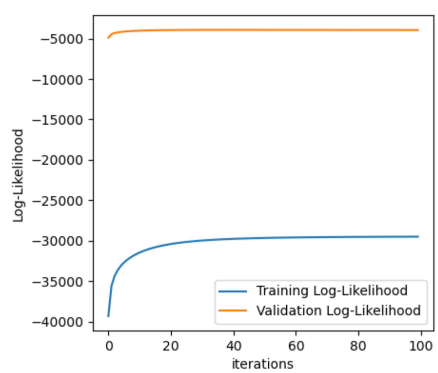

Figure 6: Training and Validation Log-Likelihoods vs Iteration

2.(c)

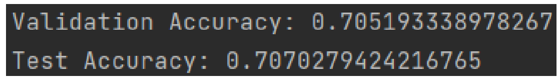
Validation Accuracy: 0.705193338978267
Test Accuracy: 0.7070279424216765

Figure 7: Final Validation & Test Accuracy

2.(d)

The three curves are all in S shape, as the sigmoid function. They represent the probability of correct responses as a function of student ability θ.
It shows that students with a high ability have a high probability of answering correctly.
Also, question with a high difficulty is skewed to the right, meaning it has a lower probability of being answered correctly.

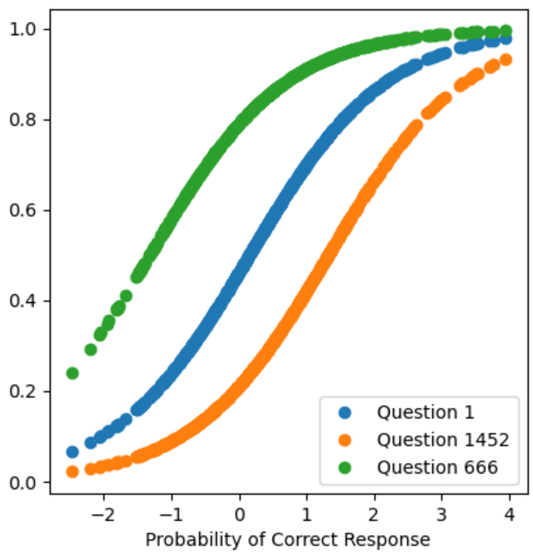

Figure 8: Probability of Correct Response vs Theta

Q3 - (i) Option 1: Matrix Factorization

3(i).(a)

```
Chosen argmax k*: 9
Validation accuracy: 0.6613039796782387
Test accuracy: 0.6587637595258256
```

Figure 9: SVD: Final Validation & Test Performance with chosen k

3(i).(b)

Filling the missing values with averages or zeros does not accurately reflect the data's underlying structure. There will be loss or distorsion of information.

3(i).(c)(d)(e)

```
Chosen argmax k*: 2,
Validation Accuracy: 0.6840248377081569
Test Accuracy: 0.6768275472763196
```

Figure 10: ALS: Final Validation & Test Performance with chosen k

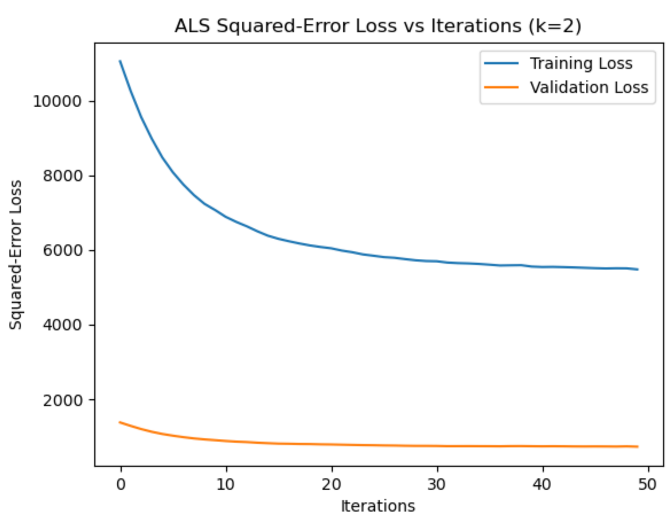

Figure 11: ALS: Squared-Error Loss vs Iterations

Q4 - Ensemble

4

Part B