CSC 311 Final Project Report

Blair Yang, Murphy Tian, Tony Chen

Due: Mar 30, 2023 at 5:00pm

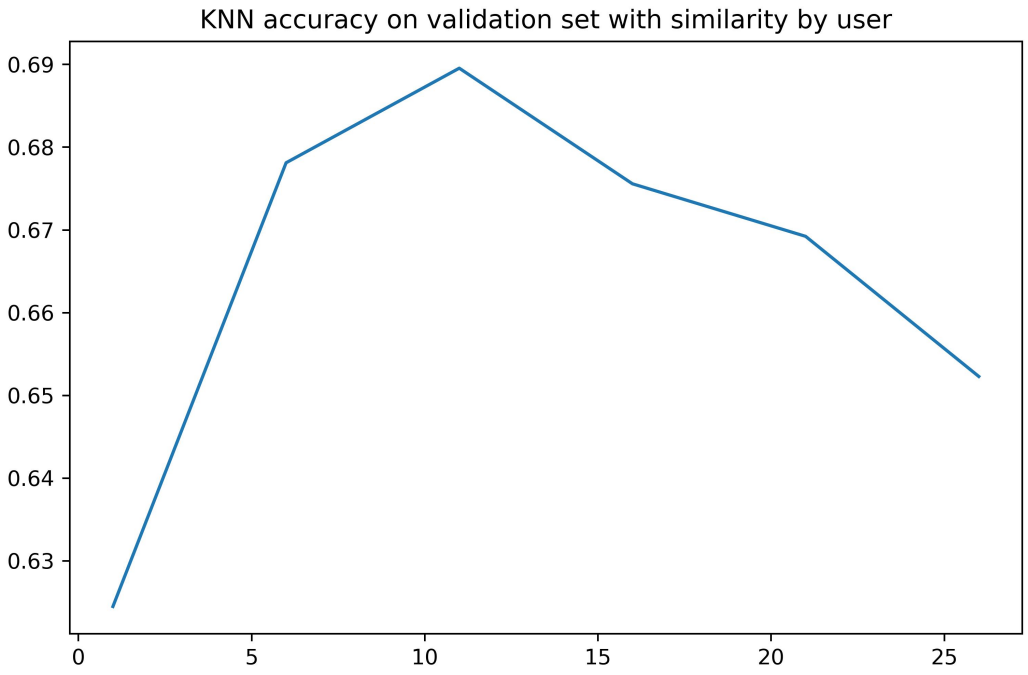
Part A

k-Nearest Neightbor

In this problem, using the provided code, you will experiment with k-Nearest Neighbor(kNN) algorithm.

The provided kNN code performs collaborative filtering that uses other students' answers to predict whether the specific student can correctly answer some diagnostic questions. In particular, the starter code implements user-based collaborative filtering: given a user, kNN finds the closest user that similarly answered other questions and predicts the correctness based on the closest students' correctness. The core underlying assumption is that if student A has the same correct and incorrect answers on other diagnostic questions as student B, A's correctness on specific diagnostic questions matches that of student B.

(a) Complete a function **main** located at **knn.py** that runs KNN for different values of $k \in \{1, 6, 11, 16, 21, 26\}$ Plot and report the accuracy on the validation data as a function of k.

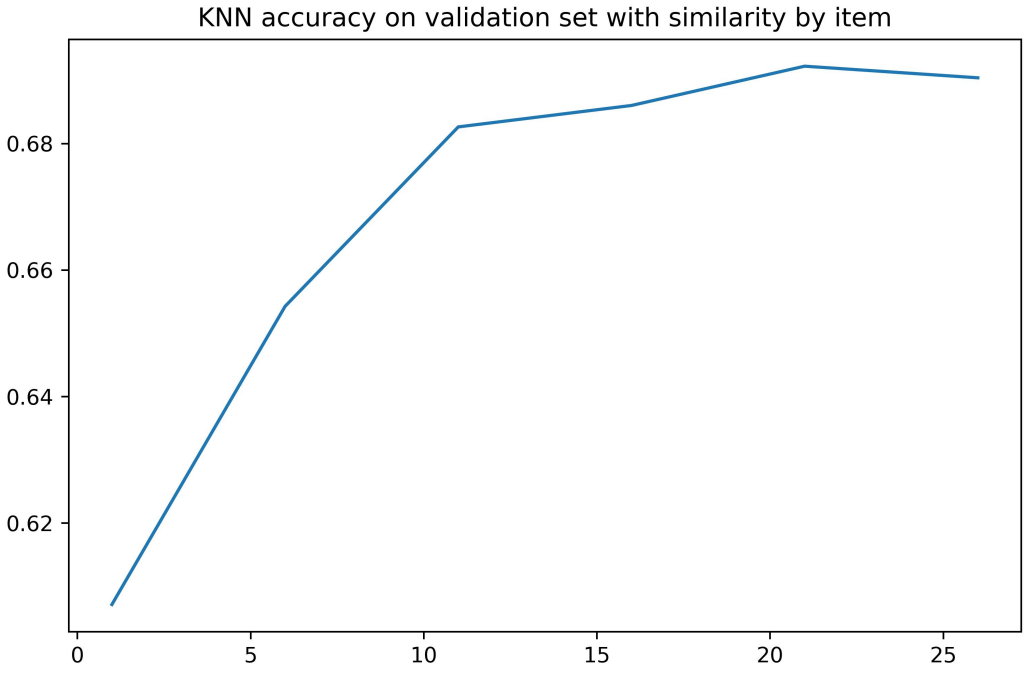


$$\text{Validation Accuracy } |_{k=1} := 0.6244707874682472$$
$$\text{Validation Accuracy } |_{k=6} := 0.6780976573525261$$
$$\text{Validation Accuracy } |_{k=11} := 0.6895286480383855$$
$$\text{Validation Accuracy } |_{k=16} := 0.6755574372001129$$
$$\text{Validation Accuracy } |_{k=21} := 0.6692068868190799$$
$$\text{Validation Accuracy } |_{k=26} := 0.6522720858029918$$

(b) Choose k^* that has the highest performance on validation data. Report the chosen k^* and the final test accuracy.

Test accuracy with maximum valid accuracy with $k^* = 11$ is 0.6841659610499576

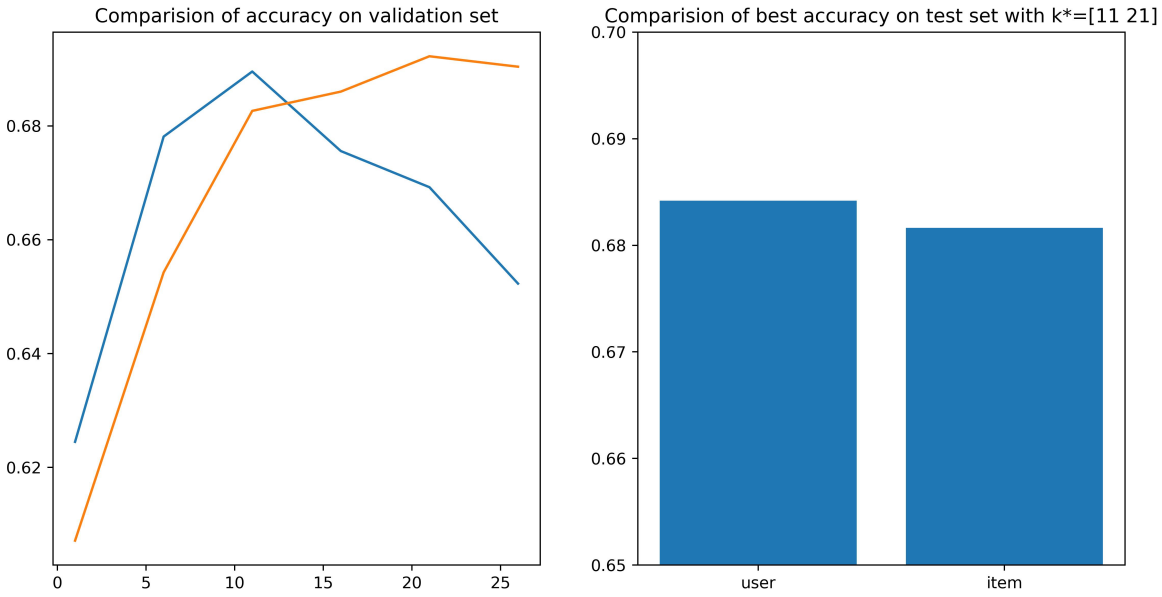
(c) Implement a function performs item-based collaborative filtering instead of user-based collaborative filtering. Given a question, kNN finds the closest question's correctness. State the underlying assumption on item-based collaborative filtering. Repeat part (a) and (b) with item-based collaborative filtering.



$$\text{Validation Accuracy } |_{k=1} := 0.607112616426757$$
$$\text{Validation Accuracy } |_{k=6} := 0.6542478125882021$$
$$\text{Validation Accuracy } |_{k=11} := 0.6826136042901496$$
$$\text{Validation Accuracy } |_{k=16} := 0.6860005644933672$$
$$\text{Validation Accuracy } |_{k=21} := 0.6922099915325995$$
$$\text{Validation Accuracy } |_{k=26} := 0.69037538808919$$

Test accuracy with maximum valid accuracy with k* = 21 is 0.6816257408975445.

(d) Compare the test performance between user- and item- based collaborative filtering. State which method performs better.



Based on the figure above, we claim them the user- collaborative filtering have better performance.

(e) List at least two potential limitations of kNN for the task you are given.

The curse of Dimensionality; bid computation cost on the test set.

Item Response Theory

In this problem, you will implement an Item-Response Theory (IRT) model to predict students' correctness to diagnostic questions.

The IRT assigns each student an ability value and each question a difficulty value to formulate a probability distribution¿ In the one-parameter IRT model, β_j represents the difficulty of the question j, and θ_i that represents the i-th students ability. Then, the probability that the question j is correctly answered by student i is formulated as:

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

a.) Derive the log-likelihood $(C|\theta, \beta)$ for all students and questions. Here **C** is the sparse matrix. Also, show the derivative of the log-likelihood with respect θ_i and β_j
(Hint: recall the derivative of the logistic model with respect to the parameters.)

From the given equation, we know that $c_{ij} = 1$ if the student answers correctly, and $c_{ij} = 0$ if the student answers incorrectly.
Hence, we claim the probability that

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

Assume the responses from students being independently and identically distributed, thus, the joint probability is the product of the individual probabilities:

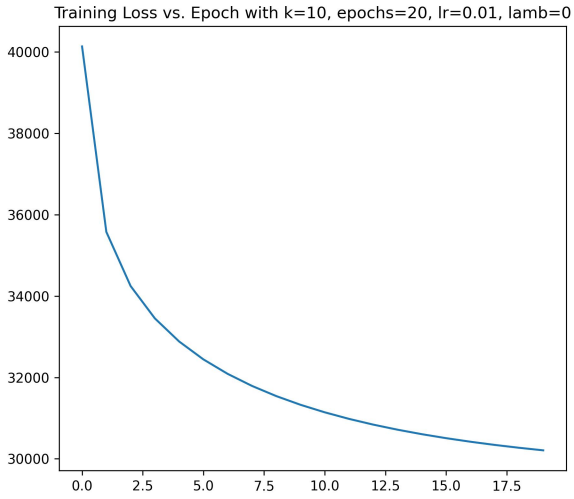
$$\begin{aligned}\log_p(C|\theta, \beta) &= \sum_i \sum_j c_{ij} \log \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)} + (1 - c_{ij}) \log \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)} \\ &= \sum_i \sum_j c_{ij}(\theta_i - \beta_j) - \log(1 + exp(\theta_i - \beta_j))\end{aligned}$$

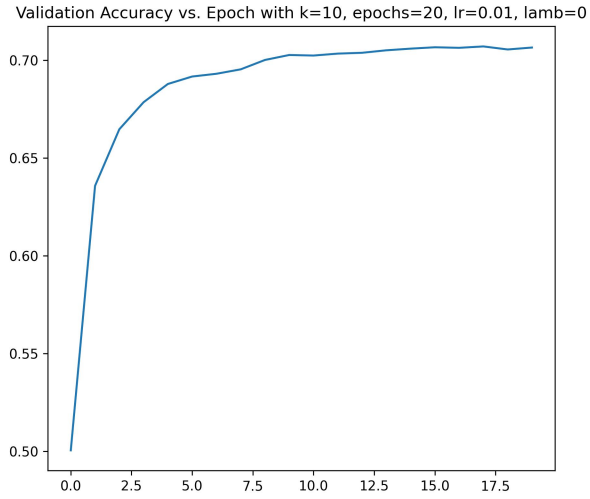
Then, taking the derivative of the logistic model with respect to θ_i, β_j, we have

$$\begin{aligned}\frac{\partial \log p(C|\theta, \beta)}{\partial \theta_i} &= \sum_j c_{ij} - \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)} \\ \frac{\partial \log p(C|\theta, \beta)}{\partial \beta_j} &= -\sum_i c_{ij} + \frac{exp(\theta_i - \beta_j)}{1 + exp(\theta_i - \beta_j)}\end{aligned}$$

b.) Implement missing functions that performs alternating gradient descent on θ and β to maximize the log-likelihood. Report the hyperparameters you selected. With your chosen hyperparameters, report the training curve that shows the training and validation log-likelihoods as a function of iteration.

Taking hyperparameter that learning rate = 0.01, iterations = 20



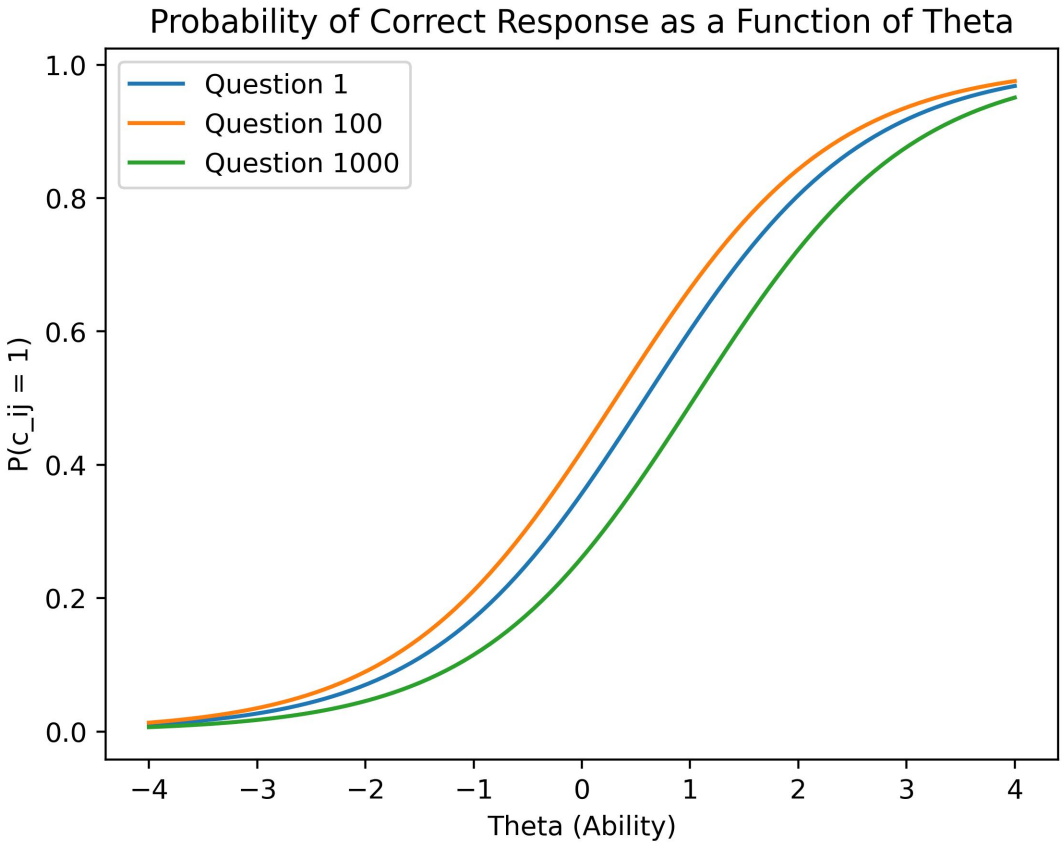


c.) With the implemented code, report the final validation and test accuracies.

the final validation := 0.709991532599492
final test accuracy := 0.7061812023708721
And detailed implementation in item_response.py file

d.) Select three questions j_1, j_2, and j_3. Using the trained θ and β, plot three curves on the same plot that shows the probability of the correct response $p(c_{ij} = 1)$ as a function of θ given a question j. Comment on the shape of the curves and briefly describe what these curves represent.

Taking questions $j_1 = 1, j_2 = 100, j_3 = 1000$.
We find that the shape of the curve is similar with a sigmoid function.
The curve on the plot shows the probability of the correct response as a function of θ given a question j.


Probability of Correct Response as a Function of Theta
Question 1
Question 100
Question 1000
P(c_ij = 1)
1.0
0.8
0.6
0.4
0.2
0.0
−4
−3
−2
−1
0
1
2
3
4
Theta (Ability)