CSCI 6517 Recommender Systems
Lab and Assignment 3: Neural Basket Complementary Recommendation

June 22, 2025

This lab and assignment involves creating a recommender engine, evaluating it to understand its performance, and making changes to improve performance where you can.

- Programming language: Python (Jupyter IPython Environment)
- Due Date: Posted in Syllabus (July 11)

Marking scheme and requirements: Full marks will be given for (1) working, readable, reasonably efficient, documented code that achieves the assignment goals and (2) for providing appropriate answers to the questions in a Jupyter notebook (named `assignment-gpt.ipynb`).

Please adhere to the collaboration policy on the course website – people you discussed the assignment solution with, or websites with source code you used should be listed in the submitted Jupyter notebook.

What/how to submit your work:

- All your plot, result tables, and answers to the questions should be included in a notebook named `assignment-gpt.ipynb`. You also need to paste 1) implementation of matrix factorization and 2) terminal outcome of your experiments into a markdown cell of the notebook.
- Submit your ipython notebook (including the PDF version of it) to BrightSpace. You must include the outputs of each cell in the submitted version. Your last submission before the assignment deadline will be considered to be your submission.

1 Before You Begin

The recommendation dataset we will be using is from Instacart, which is online grocery chain. The specific dataset we will be using for this lab is public Instacart Dataset. This anonymized dataset contains a sample of over 3 million grocery orders from more than 200,000 Instacart users.

You will also need to download the Instacart data file from kaggle website (`https://www.kaggle.com/datasets/psparks/instacart-market-basket-analysis`) and upload it into the same folder where you implement your codebase.

The weight of each question is provided on the right of each question. Total weight is- 35.

Should you need to create new cells in the notebook, you can do so. Also, there are some codes commented out to help you save and load data, and to run in Google Colab. Uncomment if you are using these. Complete all sections of the lab notebook.

2 Main Assignment

Please answer the questions below and provide IPython implementations.

Objective: You are going to use a four-layer GPT architecture to train an sequential recommender system that serves for within-basket recommendation tasks. Specifically, given a small set of items (say 4 items) are already in a shopping cart, your goal is to "complete" a shopping cart by suggesting more items that complement the items in the cart.

Useful Information:

1. GPT implementation tutorial.
   - Example ipython notebook code
   - Andrej Karpathy' GPT video tutorial

   Hint: once you understand the input and output of the GPT model, the only remaining thing is to feed recommendation dataset into it in a right format and train.
2. Recommender system evaluation code. This is the same code you used in your assignment 2.
3. Remove "order.csv" file (you don't need it). Treat each order as a "user". Your goal is to "complete" a shopping cart (order) given a small set of items are already in the cart. E.g. a cart contains coffee cream powder, what would be the next time the user may want to buy?
4. Remove "order_products__prior.csv" (you don't need it). It is too big for you to complete the assignment.

Q1. Data Preprocessing

(a) Given the Instacart dataset (order_products__train.csv), split dataset based on order ids. 70% orders as **holdout** set and 30% orders as **cold-start** test set.

- Find out how many groups are there in holdout and cold-start groups. (3 pt)

- On average, how many items in each basket in holdout and cold start groups respectively? (3 pt)
- How many groups are there with more than 8 items? (2 pt)

(b) For your holdout dataset obtained from previous step, for each order, put first 8 products (items) into **training** set and put the rest into the **validation** set. If the cart does not have more than 8 items, remove the order from your data pool. Answer the following questions

- How many orders are in your holdout and cold-start datasets after removing invalid orders? (2 pt)
- Is there any repeated purchase in each of the orders? (2 pt)
- Print the top 5 orders of your training dataset as a list of lists. (2 pt)

Q2. Train GPT as a sequential recommender system

(a) Understand GPT architecture and training schema. Answer the following questions:

- If a GPT model is trained on input sequences with the length of 8, is it possible for it to take a sequence of length 9 as input and make a prediction? Why? (2 pt)
- Why do we need position encoding when using the GPT model? Do you need to have position encoding for within-basket recommendation task? Why? (2 pt)

(b) Use training set to train your GPT model. Show the training curve as a plot. (2 pt)

Q3. Performance Evaluation

(a) Implement performance evaluation code for your GPT-based sequential recommender system.

(b) Evaluate the model's performance in 2 scenarios

- For each cart (order) in the holdout dataset, take its 4th to 8th products as a input sequence, feed them into the GPT-based sequential recommender system, and make prediction (using output gate 4). Evaluate how well the model performs in terms of matching the remaining items in the cart. Report Recall@10, Precision@10, and NDCG@10. (5 pt)
- (Challenge!) For each cart (order) in the cold-start dataset, take its first four products as input and feed them into the GPT-based sequential recommender system. Evaluate how well the model performs in terms of matching the remaining items in the cart. Report Recall@10, Precision@10, and NDCG@10. (2 pt)

(c) Answer the following questions:

- Does the model perform well (in terms of Recall)? Why? (3 pt)
- Can you do any modifications that can improve the model performance? If so, please describe your changes, report the improved performance, and justify why it works. (2 pt)