CSCI 1952-Q: Algorithmic Aspects of Machine Learning (Spring 2023)

Coding Assignment 1

Due at 2:30pm ET, Tuesday, Mar 7

Getting Started.

- You are free to use any programming language of your choice.
- You are encouraged to discuss with other students and you can use resources on the web, but you must write your code independently. You should acknowledge who you discussed with and which resources you used.

Assignment Overview. In this assignment, you will try to build a recommendation system. You will be given as input different users' ratings for different movies, obtained from a real-world dataset. Based on the provided ratings, you need to predict how certain users will rate certain movies (that are not part of the input). Your goal is to minimize the error of the predicted ratings. You are free to use any other algorithms to make these predictions.

Input. The input has two parts: (1) the training data, consisting of k ratings across m movies and n users, and (2) the q queries for which your algorithm needs to provide predictions.

You are given an input file `mat_comp`. The first line of this file contains three integers n, m, k. This is followed by k lines, where each line contains three numbers i, j, and $M_{i,j}$, specifying user i's rating for movie j. The ratings are made on a 5-star scale with half-star increments ($0.5 \leq M_{i,j} \leq 5.0$). On the next line, the input file specifies an integer q. This is followed by q lines, where each line contains two integers i, j, asking you to predict how user i will rate movie j.

Output. The output file should have q lines with a single real number on each line, reporting your algorithm's predictions for the q queries.

Submission.

- Your submission should consist of exactly 3 files:
  1. an output file `mat_comp_ans` in the specified format,
  2. a well-commented source code in a language of your choice, and
  3. a `pdf` file containing a detailed explanation of your approach (e.g., an overview of your approach, what you did to achieve the final solution).
- We may ask you to show us that running the submitted code does produce the submitted output file.

- In addition to the input file `mat_comp`, we also provide a smaller input file `mat_comp_small`, which has the same format. You should work with one input file and submit exactly one output file. You should name your output file `mat_comp_ans` or `mat_comp_small_ans` accordingly. In short, you should run your code on `mat_comp` if your code can handle it, otherwise you can run in on `mat_comp_small`.

Test Loss. Let S denote the set of entries in the test data. Suppose $M_{i,j}$ are the actual ratings (hidden from you) and $A_{i,j}$ are the ratings you predicted. We define the test loss to be:

$$L = \frac{1}{|S|} \sum_{i,j \in S} \left(M_{i,j} - A_{i,j} \right)^2 \ .$$

Grading. This assignment will be graded out of 6 points:

- (2 points) Code readability and the explanation of your approach.
- (4 points) You will get a score of $4 \cdot (1.8 - L)$ where L is the test loss defined earlier. If the score is lower than 0 or higher than 4, it is set to 0 or 4. In particular, you will receive full credit if your test loss is 0.8 or lower.
- (1 bonus point) you will receive 1 bonus point if your test loss is among the smallest 20% of all received submissions.
- You will lose 1 point if you work with the small input file.

Dataset. The input files were obtained from the `ml-latest` MovieLens dataset available at `https://grouplens.org/datasets/movielens/` [HK16]. The dataset describes 5-star rating from the movie recommendation service MovieLens (`http://movielens.org`). The `ml-latest` dataset contains 27753444 ratings across 58098 movies, created by 283228 users between January 09, 1995 and September 26, 2018. The usage license for this dataset is specified on the GroupLens website.

For this coding assignment, we trimmed the `ml-latest` dataset as follows. We repeatedly removed users who rated fewer than 497 movies and movies that has fewer than 71 ratings. After this trimming process, the user-movie rating matrix contains 10104 users, 10234 movies, and 8882248 ratings. A random (roughly) 10% of these ratings are kept secret from you as test data, while the remaining 90% of the ratings are given to you as training data.

The small input file is then obtained by further randomly retaining 10% of the users and movies, resulting in 1010 users and 1023 movies. The two input files (e.g., the train/test split) are otherwise identical.

Remarks/Hints. While the intended solution is to use non-convex algorithms for low-rank matrix completion, this is optional. You are free to use any other algorithms to make these predictions. Due to this reason, the following hints may not apply to your solution.

- Even though you do not have access to the test data, you can use cross-validation to estimate the performance of your code (and for tuning hyper-parameters).
- Let $M \in \mathbb{R}^{n \times m}$. A natural non-convex objective for asymmetric matrix completion is

$$f(X,Y) = \sum_{i,j \in \Omega} \left(M_{i,j} - (XY^\top)_{i,j} \right)^2$$

where $X \in \mathbb{R}^{n \times r}$, $Y \in \mathbb{R}^{m \times r}$, and Ω is the set of observed entries. (If you are using this approach, you need to choose the rank yourself.)

- One could initialize X and Y using the SVD of M (with all unknown entries set to 0).
- One could use stochastic gradient descent, where in each iteration only one row of X and one row of Y is updated.
- One could add regularizers to this objective function. For some common choices see, e.g., Section 3.2 of [GJZ17].

References

[GJZ17] R. Ge, C. Jin, and Y. Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 1233–1242. PMLR, 2017.

[HK16] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, 2016.