CSCI 1952Q: Algorithmic Aspects of Machine Learning (Spring 2024)

Coding Assignment 1

Due at 1:00pm ET, Friday, Feb 23

Getting Started.

- You can use any programming language for this coding assignment.
- You cannot use packages or functions that directly solve the problem.

Overview. In this assignment, you will build a key component of a recommendation system. You will be given as input various users' ratings for different movies, sampled from a real-world dataset. Your task is to predict how certain users will rate certain movies based on the provided ratings. The objective is to minimize the error of your predicted ratings. You are free to use any 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 will be given an input file `mat_comp`. The first line of this file has three integers: n, m, and k. This is followed by k lines, each containing three numbers i, j, and $M_{i,j}$, specifying user i's rating of movie j. The ratings are on a 5-star scale with half-star increments ($0.5 \le M_{i,j} \le 5.0$). The next line has an integer q. This is followed by q lines, each containing two integers i and j, asking you to predict how user i will rate movie j.

Output. The output file should have q lines, each containing a single number. These are your algorithm's predictions for the q queries (which do not have to be in increments of 0.5).

Submission.

- Your submission should consist of exactly 3 files:
  1. An output file `mat_comp_ans` in the specified format.
  2. A text file (e.g., `.cpp`, `.py`) containing your source code.
  3. A `.pdf` file providing a detailed explanation of your approach.
- We may ask you to show us that running the submitted code does produce the submitted output file.

Evaluation. Let S denote the set of queried entries. Suppose $M_{i,j}$ are the actual ratings (hidden from you) and $A_{i,j}$ are the ratings you predicted. The test loss is defined as:

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

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

- (1 point) Your code should have good readability and should be well commented.
- (1 point) Your explanation `pdf` must be typed (e.g., MS Word or LaTeX). You should give an overview of your ideas and approach in the first 2 pages. Material beyond the first 2 pages will be read at the discretion of the instructor/TAs.
- (4 points) You will receive a score of $(7.2 - 4L)$ where L is your test loss. If your score is lower than 0 or higher than 4, it will be 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.
- We may deduct up to 4 points for any formatting error in your output (including but not limited to: not naming the output file `mat_comp_ans`, not outputting exactly q lines, or not outputting a single real number in each line).

Dataset. The input data is based on the `ml-latest` MovieLens dataset [HK16]. [1] As of Feb 2024, this dataset has 33832162 ratings across 86537 movies, provided by 330975 users between 1995 and 2023 on the movie recommendation service MovieLens. [2] The usage license for this dataset is specified on the GroupLens website.

For this coding assignment, we iteratively remove users who rated fewer than 200 (remaining) movies and movies that was rated by fewer than 50 (remaining) users. After this trimming process, there are 42662 users and 15304 movies. Next we sample 20000 users and 5000 movies randomly and discard the rest. After this sampling, we have 3187989 ratings. A random (roughly) 10% of these ratings are withheld as test data, while the remaining 90% are given to you as training data.

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

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

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

  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, the rank r is a hyperparameter that you need to choose.)
- One could initialize X and Y using the SVD of M (with all unknown entries set to 0).
- One could use stochastic gradient descent, updating only one row of X and one row of Y in each iteration.
- One could add regularizers to the objective function. Some possible options are discussed in, e.g., Section 3.2 of [GJZ17].

[1] Available at `https://grouplens.org/datasets/movielens/`.

[2] `http://movielens.org`.

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.