

CSCI 1520: Algorithmic Aspects of Machine Learning (Spring 2025)

Coding Assignment 3

Due at 11:59pm ET, Thursday, April 24

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 core component of a recommendation system. You will be given various users' ratings for different movies, sampled from a real-world dataset. Your task is to predict how certain users will rate certain movies and minimize the prediction error.

Input. You will be given an input file `movie_ratings`, which contains k known ratings between n users and m movies, followed by q user-movie pairs for you to predict. 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 ($1 \leq i \leq n$ and $1 \leq j \leq m$). The ratings are on a 5-star scale with half-star increments ($0.5 \leq M_{i,j} \leq 5.0$). 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 predictions for the q queries. The predicted ratings do not have to be in increments of 0.5.

Submission.

- Your submission should consist of exactly 3 files:
 1. An output file `matcomp_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 be the set of user-movie pairs that you are asked to predict. Let $M_{i,j}$ denote the ground-truth (hidden) ratings. Let $A_{i,j}$ denote your predictions. The test loss is defined as:

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

The autograder will provide your average loss on (a randomly chosen) 1% of S as feedback.

Grading. This assignment will be graded out of 14 points:

- (3 points) Your code should have good readability and should be well commented.
- (3 points) 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.
- (8 points) You will receive a score of $(14.4 - 8L)$ where L is your test loss. If the score is lower than 0 or higher than 8, it will be set to 0 or 8. In particular, you will receive full credit if $L \leq 0.8$.
- (2 bonus points) You will receive 2 bonus points if your test loss is among the smallest 20% of all received submissions.
- We may deduct up to 8 points for any formatting error in your output (including but not limited to, not naming the output file `matcomp_ans`, not outputting exactly q lines, or not outputting a single real number in each line).

Dataset. The input data was obtained from the MovieLens dataset [HK16].¹ Specifically, the `ml-latest` dataset generated in July 2023 was used. This dataset contains 33832162 ratings for 86537 movies, provided by 330975 users between 1995 and 2023 on the MovieLens website.² The usage license for this dataset is specified on the GroupLens website.

For this coding assignment, we first iteratively remove users who rated few movies and movies rated by few users. Then, we randomly sample a subset of 20000 users and 5000 movies and discard the rest. We have 1582018 ratings between these users and movies. A random (roughly) 10% of these ratings are withheld as test data, and the remaining 90% are given to you as training data.

Hints. You are not required to use low-rank matrix completion, so the following hints may not be relevant to your approach. You are encouraged to explore other algorithms to solve this problem.

- It is recommended to use cross-validation to evaluate the performance of your code and tune hyperparameters.
- Let $M \in \mathbb{R}^{n \times m}$. A natural non-convex objective for asymmetric matrix completion is

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

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 use 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].

¹Available 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.