Machine Learning & Data Mining
Miniproject 2: MovieLens Visualization

Caltech CS/CNS/EE 155
Released February 21^{st}, 2025

---

1 Overview

- In this project, you will be creating visualizations of the MovieLens data set using matrix factorization.
- Each team must post at least one interesting visualization and a brief discussion of it on Piazza by **11:59pm on Friday, February** $28^{\textbf{th}}$. Report due by **11:59pm on Saturday, March** $1^{\textbf{th}}$, via Gradescope.
- Submit your report as a single .pdf file to Gradescope, under "MiniProject 2". There is no fixed formatting for this report. Once again, please submit your report in groups rather than submitting it once per student.
- Submit your code by **sharing a link in your report** to your Google Colab notebook for each problem. Make sure to set sharing permissions to at least "Anyone with the link can view". **Links that can not be run by TAs will not be counted as turned in.** Check your links in an incognito window before submitting to be sure.
- Be sure to appropriately title all visualizations wherever they appear so that it is clear which question each visualization corresponds to.
- When accessing the MovieLens data from your colab notebook, please do so via a link from the raw github data in the course github.
- You should work in a group of size 2 to 4. We encourage you to use the Search for Teammates feature on Piazza to help you find teammates. You may keep the same group as in the previous miniproject.
- You may collaborate fully within your team, but not collaboration is allowed between teams.

2 LLM Usage

This section has an example of LLM usage reporting below. Please follow this format to report LLM usage. Indicate each usage clearly.

- Name of LLM(s) Used: ChatGPT
- Components of Project involving LLM: Data processing.

3 Background

In late 2006, Netflix challenged the world to create a recommender system that could predict whether a user would like a given movie based on his/her previous ratings on other movies. Netflix created their own recommender system, Cinematch, and hoped that the world could beat their performance by over 10% (in terms of how closely predicted ratings match subsequent actual ratings). The challenge ended in September 2009, when team "BellKor's Pragmatic Chaos" surpassed the 10% mark.

In this miniproject, we will focus on creating visualizations of this data, rather than predicting user ratings on movies. In order to reduce the computational time needed to produce these visualizations, we will be working with the smaller MovieLens Dataset rather than the full Netflix Prize Dataset. We will start with some basic visualizations and then move on to more complicated ones.

4 Data Format

The MovieLens data set consists of 100,000 ratings from 943 users on 1682 movies, where each user has rated at least 20 movies. More information about the files can be found below:

- **movies.csv**: Each of the 1682 lines in this file contains a comma-delimited list of the following fields for a movie:

  Movie ID, Movie Title, Unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western

  The last 19 fields are various movie genres. Here, a 1 indicates the movie is of the given genre, while a 0 indicates that it is not. Note that movies can be in several genres at once. The movie ids correspond to the movie ids specified in the **data.csv** file and range from 1 to 1682.

  https://caltech-cs155.s3.us-east-2.amazonaws.com/sets/miniprojects/project2/data/data.csv

- **data.csv**: Each of the 100,000 lines in this file consists of a comma-delimited list of the following fields for a given rating instance:

  User ID, Movie ID, Rating

  Here, all ratings are integer values ranging from 1 to 5. User ids range from 1 to 943 and movie ids range from 1 to 1682, as in the previous file.

  https://caltech-cs155.s3.us-east-2.amazonaws.com/sets/miniprojects/project2/data/data.csv

5 Report Guidelines

If your team uses a Large Language Model (LLM) for any part of this competition, you must report the usage. In the report, clearly show the prompt used and the output given. An example of this is given in the LLM Usage section of the template file.

Introduction [5 points]

Include an introduction page that contains your team name, a list of all team member names, the work division, and the packages used for this project.

Basic Visualizations [20 points]

First, you will create some basic visualizations of the MovieLens dataset described above. Using a method (e.g. histograms) of your choice, visualize the following:

1. All ratings in the MovieLens Dataset.
2. All ratings of the ten most popular movies (movies which have received the most ratings).
3. All ratings of the ten best movies (movies with the highest average ratings).
4. All ratings of movies from three genres of your choice (create three separate visualizations).

Note that in Parts 2 and 3 you only need to make one combined histogram for the ten most popular movies and one combined histogram for the ten best movies.

The Python packages Matplotlib and Seaborn are good choices for these visualizations, but there are also many other good visualization packages.

Report Deliverable

Your report should contain a section dedicated to basic visualizations. What, in general, did you observe? Did the results match what you would expect to see? How do the ratings from the most popular movies compare to the ratings of the best movies? How do the ratings of the three genres you chose compare to one another?

Matrix Factorization Visualizations [60 points]

Let m, n be the number of users and movies, respectively, and Y be the m x n matrix of the movie ratings, where y_{ij} corresponds to user i's rating for the movie j. Note that most of the elements of the matrix are unknown. The goal of a recommender system is to predict these missing values.

Your job is to find the matrices U and V, such that $Y \simeq U^T V$. Note that U has dimension $k \times m$ and V has dimension $k \times n$. You must try the following three methods for finding U and V.

1. Use (and/or modify) your code or the solution code for Homework 5.
2. Incorporate bias terms a and b for each user and movie, to model global tendencies of the various users and movies. See the guide for more information.
3. Use an off-the-shelf implementation[1]. You can Google "collaborative filtering python," etc. to see examples. Note that in this assignment, we want you to try an off-the-shelf matrix factorization method, rather than any collaborative filtering method in general.

[1] One off-the-shelf method we suggest is Surprise SVD, found at http://surpriselib.com/.

For the first two methods, choose *k = 20*, and justify your choices for any other parameters and the stopping criteria you use. For all of these methods, split the MovieLens dataset into a training set (of size 90,000) and a test set (of size 10,000), as given in the files **train.csv** and **test.csv**. You should then compare these methods by assessing their performance on the test set. Once you have obtained U, V, you will attempt to visualize and interpret your results.

1. In order to visualize the resulting latent factors, apply SVD to $V = A\Sigma B$ and use the first two columns of A to project *U*, *V* into a two-dimensional space. This projection is given by $\tilde{U} = A_{1:2}^T U \in \mathbb{R}^{2\times m}$ and $\tilde{V} = A_{1:2}^T V \in \mathbb{R}^{2\times n}$.
2. Now, construct creative 2D-visualizations of $\tilde{V}$, similar to the one in Figure 2 of the reference [1]. Visualize the following:
   (a) Any ten movies of your choice from the MovieLens dataset.
   (b) The ten most popular movies (movies which have received the most ratings).
   (c) The ten best movies (movies with the highest average ratings).
   (d) Ten movies from the three genres you selected in Section 4, Basic Visualizations (for a total of 30 movies). Create one visualization, containing ten movies, for each of the three genres you select.

You should include 6 graphs for each of the 3 different types of SVD that you implemented, for a total of 18 graphs.

Report Deliverable

Your report should contain a section dedicated to matrix factorization methods. How do each of these methods work? Please be specific and include equations. How do they differ? How did they perform in comparison to one another on the test set? Can these methods' differences explain why they perform differently on the test set?

Your report should also contain a section dedicated to matrix factorization visualizations. What, in general, did you observe? Did the results match what you would expect to see? How does the visualization of the most popular movies compare to the visualization of the best movies? How do the visualizations of the three genres you chose compare to one another? How do the visualizations produced by the different matrix factorization methods compare to one another? Be sure to include some plots to indicate which phenomena you're referring to with respect to your observations.

Piazza Post [15 points]

In addition to submitting the project on Gradescope, one group member of each team should make a post on Piazza at the aforementioned date and time of at least one interesting visualization the team created. The post should be made in the project2 folder with the title "[Team Name]: Visualization Submission," and the first line of the post should be "Submitted by: [Team Members]." The post should contain at least one visualization created using matrix factorization and a brief description (1 paragraph should suffice) of what makes the visualization so interesting. We will put a sample post on Piazza for reference.

References

1. Koren, Y., Bell, R.,& Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems Computer, (8), 30-37.

2. Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 230-237). ACM.