Comp 135 Intro ML: Project 3

Tufts 2018 Spring, Instructor: Liping Liu

Due date: 04/20, 11:59:59pm

1 Introduction

In this project, you are asked to build a recommendation system. You are given a dataset of movie ratings by users. Based on ratings you have, you need to recommend five movies to each user, and your recommendations are evaluated on our held-out dataset.

Associated with the ratings, you also have some information about users and movies. For example, user ages, user genders, and release years of movies. You will also need to analyze your learned model with such data.

2 Collaborative Filtering

In this problem, you are asked to run collaborative filtering to make recommendations.

As we have talked in the class, we can find vector representations of users and items through SVD decomposition of the incomplete matrix of ratings. The training objective is

$$\min_{(\mathbf{u}_i)_{i=1}^N, (\mathbf{v}_j)_{j=1}^L} \quad \sum_{i=1}^{N}\sum_{j=1}^{L}(M_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2 + \lambda \sum_{i=1}^{N} \|\mathbf{u}_i\|^2 + +\lambda \sum_{j=1}^{L} \|\mathbf{v}_j\|^2$$

Here each user i and each movie j are represented by $\mathbf{u}_i \in \mathbb{R}^K$ and $\mathbf{v}_j \in \mathbb{R}^K$ for . Essentially we are using low dimensional vectors to represent users and items.

You can use other variants of collaborative filtering algorithms to make recommendation. You can either write your own code to solve the problem or use existing python packages (e.g. `surprise`).

You need to use the model selection procedure to decide the value of model hyperparameters (λ and K in the formulation above). One way of doing so is to hold out some ratings as the validation set.

3 Result Analysis

In this project, you are asked to analyze the learned model to get better understanding of collaborative filtering. Especially, you are asked to look into learned vectors and see if there is any meaningful information.

We first compare user vectors and user information. Let $\mathbf{U}$ be the matrix of user vectors, with each row as the vector of each user. Each column of $\mathbf{U}$ can be seen as a "learned" feature of users. Now we want to check whether any of these columns are meaningful. We compare every column

U的两张图：user age/ user gender
V一张图：release year

of **U** with user information, e.g. user ages, by calculating correlation coefficient of the two. If one column has strong correlation with user ages, then we say the model has learned some information about user age from user ratings of movies.

In this task, you are asked to calculate correlation coefficient between each column of **U** and user ages, find the column with the largest correlation coefficient with user ages, and plot values in that column against user ages. Then do the same thing to make a plot of one column of **U** and user genders.

In the second step, you are asked to compare movie vectors and release years of movies. Let **V** be a matrix with each row as a movie vector. You need to generate a plot of one column of **V** and release years in the same way as you do for users.

4 Submission requirements

You need to follow submission requirements in this section to guarantee that you submission can be evaluated properly.

In your submission, you need to include 1) a zip file containing your code, 2) a result file containing your recommendations, 3) your nickname in a file, and 4) your report.

Your submitted code should clearly show your model training process. You should write clear documentation for your code.

In your result file, you should recommend five movies for each user. The file should be in the following format.

```
user_id, rec1, rec2, rec3, rec4, rec5
0, 2, 1, 3, 5, 9
1, 12, 3, 1, 9, 6
...
```

Your result will be evaluated by the mean rating of your recommended movies. We calculating the mean rating in two ways depending on how we assign ratings to the recommended movies not in our held-out data. In the first way, we assign 0 to such a movie, meaning that the user does not like the movie if the user does not watch the movie in the held-out data. In the second way, we assign 2 to such a movie, meaning that we make a rough estimation of the user rating of the movie when the user does not actually rate the movie.

As we will generate a leaderboard for the competition, we need your nicknames to show the performance of your classifiers. You may want a cool nickname but others cannot identify you from it.

Your report should clearly states what you have done for the project.

- State the learning algorithms you have used and the parameter setting of your algorithm. You should have the model selection procedure to decide model hyperparameters. Your report should be clear enough for one to repeat your experiment by only reading your report.
- Report your analysis of learned vectors. Include the plots required in Section 3 in your report.
- Write a short paragraph to discuss how to incorporate user information and movie information into the recommendation model.

You are encouraged to describe any interesting findings from this project. Try to make the report pleasant to read – not only dry results.

5 Evaluation

In this project, your submission is evaluated based on your code, your analysis of the learned model, and your report.

- You get 40% of full points if you have correctly get a result of recommendations.
- You get 40% of full points if you have analyzed user vectors and movie vectors described in Section 3.
- You get 20% of full points if your report clearly shows what you have done and satisfies the requirements.

If you do not get full points for any grading item, you get partial points according to our estimation of the amount of your effort.

6 The data

The data is from MovieLens. There are 83115 ratings by 943 users on 1682 movies. All the data you need are in three files, `u.data`, `u.user`, and `u.item`. The first file contains ratings of movies from users, the second file contains information about users, and the third file has information about movies. You will find detailed information about these files in a `README` file.