CSE 258, Fall 2017: Homework 3

Instructions

Please submit your solution **by the beginning of the week 7 lecture (Nov 13).** Submissions should be made on **gradescope**. Please complete homework **individually**.

These homework exercises are intended to help you get started on potential solutions to Assignment 1. We'll work directly with the Assignment 1 dataset to complete them, which is available here:
`http://jmcauley.ucsd.edu/data/assignment1.tar.gz`

Executing the code requires a working install of Python 2.7 or Python 3.

You'll probably want to implement your solution by **modifying the baseline code provided**.

Note that you should be able to join the competitions using a UCSD e-mail. The competition pages can be found here:
`https://inclass.kaggle.com/c/cse158-258-fa17-visit-prediction`
`https://inclass.kaggle.com/c/cse158-fa17-categorization`
`https://inclass.kaggle.com/c/cse258-fa17-rating-prediction`
though you will need to use the login keys to access the competition:
visit prediction: `https://www.kaggle.com/t/f5cb34fc0265469eb7c4d4e86618e650`
rating prediction: `https://www.kaggle.com/t/a9061008f33740a6a0b8e120b775bc7b`

Please include the code of (the important parts of) your solutions.

Tasks (Visit prediction)

First, since the data is quite large, when prototyping solutions it may be too time-consuming to work with all of the training examples. Also, since we don't have access to the test labels, we'll need to simulate validation/test sets of our own.

So, let's split the training data ('train.json.gz') as follows:
(1) Reviews 1-100,000 for training
(2) Reviews 100,001-200,000 for validation
(3) Upload to Kaggle for testing only when you have a good model on the validation set. This will save you time (since Kaggle can take several minutes to return results), and also will stop us from crashing their website...

1. Although we have built a validation set, it only consists of positive samples. For this task we also need examples of user/business pairs that *weren't* visited. Build such a set by randomly sampling users and businesses until you have 100,000 non-visited user/business pairs. This random sample combined with your 100,000 validation reviews now corresponds to the complete validation set for the visit prediction task. Evaluate the performance (accuracy) of the baseline model on the validation set you have built (1 mark).

2. The existing 'visit prediction' baseline just returns *True* if the business in question is 'popular,' using a threshold of the 50th percentile of popularity (`totalVisits/2`). Assuming that the 'non-visited' test examples are a random sample of user-visit pairs, is this particular threshold value the best? If not, see if you can find a better one (and report its performance), or if so, explain why it is the best (1 mark).

3. Users may tend to repeatedly visit business of the same type. Build a baseline that returns 'True' if a user has visited a business of the same category before (at least one category in common), or zero otherwise (1 mark).[1]

4. To run our model on the *test* set, we'll have to use the files 'pairs_Visit.txt' to find the userID/businessID pairs about which we have to make predictions. Using that data, run the above model and upload your solution to Kaggle. Tell us your Kaggle user name (1 mark). If you've already uploaded a better solution to Kaggle, that's fine too!

Tasks (Rating prediction)

Let's start by building our training/validation sets much as we did for the first task. This time building a validation set is more straightforward, you can simply use half of the data for validation, and do not need to randomly sample non-visited users/businesses.

[1]Here consider the 'category of a business' to be the set of all categories users selected for that business.

5. What is the performance of a trivial predictor

$$\text{rating}(\text{user}, \text{item}) = \alpha$$

on the validation set, and what is the value of α (1 mark)?

6. Fit a predictor of the form

$$\text{rating}(\text{user}, \text{item}) \simeq \alpha + \beta_{\text{user}} + \beta_{\text{item}},$$

by fitting the mean and the two bias terms as described in the lecture notes. Use a regularization parameter of $\lambda = 1$. Report the MSE on the validation set (1 mark).

7. Report the user and item IDs that have the largest and smallest values of β (1 mark).

8. Find a better value of λ using your validation set. Report the value you chose, its MSE, and upload your solution to Kaggle by running it on the test data (1 mark).