Advanced Machine Learning for Personalization 6998-2
Homework 1
Due February 23, 2018 before 10am.

You will build a movie recommendation engine that performs collaborative filtering by using the MovieLens 20M dataset. The data-set contains 20000263 ratings across 27278 movies as generated by 138493 users between January 09, 1995 to March 31, 2015. All selected users have rated at least 20 movies.

Start by downloading the data-set at

`http://files.grouplens.org/datasets/movielens/ml-20m.zip`

Unzip the download and check that you have the necessary .csv files. The key file of interest is ratings.csv which contains the ratings some users gave to some movies on a 5-star scale with half-increments (e.g. 0.5 stars to 5.0 stars). On each line, the file has a rating with the following format

$$(userId, movieId, rating, timestamp)$$

If you want additional information about the movies (though it is not necessary for this homework), you can find it in the file movies.csv which has a row per movie containing the $(movieId, title, genres)$ where genres is a pipe-separated list describing what categories the movie is in (e.g. Action, Adventure, etc.). Feel free to ignore this extra information as it is not necessary for this exercise.

You will implement Matrix Factorization and Matrix Completion by decomposing the matrix into a low rank factorization. Since you do not know the rank of the matrix factorization in advance, you must sweep across various values of the rank r to see where performace for Matrix Completion is best. Split the ratings.csv data randomly into half to form a train Ω and test Ω_{test} split. Do so by sampling half of each user's rated movies in training and the other half as testing. That way, you see each user in your training data and in your test data (this makes the problem easier). Train your matrix factorization for various ranks r by minimizing (e.g. via gradient descent or some other variation if you prefer) the following cost function

$$\min_{V,W} \sum_{(i,j)\in\Omega} (R_{i,j} - (VW^\top)_{i,j})^2 + \lambda(\|V\|_F^2 + \|W\|_F^2)$$

subject to the constraint that the rank of V and W are set to r for various values of r and for various values of the regularizer λ. Once converged, report the RMSE and the MRR of your factorization on the held out Ω_{test} across multiple r values and λ values. Recall that RMSE is simply

$$RMSE = \sqrt{\sum_{(i,j)\in\Omega_{test}} (R_{i,j} - (VW^\top)_{i,j})^2}.$$

To compute MRR, for each user, compute a ranking over the movies that were not in the training data for the user. Then, use your ranking to see how well you would retrieve the remaining movies that user u rated in Ω_{test} which had a rank that is larger than or equal to 3 stars. Call that set of unseen movies Ω_u for user u. Then compute the MRR for user u as follows

$$MRR_u = \frac{1}{|\Omega_u|} \sum_{i \in \Omega_u} \frac{1}{rank_i}$$

using the ranking for the user that you obtained from your decomposition. Then output the average over MRR_u across all users for that particular choice of r and λ.

Do this for the range of r values and λ values. Write a report describing your work, your results. For instance, show the plots of RMSE and MRR across r and λ for training and test data (ideally as a surface plot). Make sure to run the code over multiple random folds in order to get the average RMSE and average MRR across r values (as well as another plot for the standard deviation).

Upload all work to courseworks.columbia.edu. You can use ANY language to implement this homework. Please organize your code into separate files when appropriate. Your write-up should be in Adobe Portable Document Format (.pdf). Please do not submit Microsoft Office documents, LaTeX source code, or something more exotic since we will not be able to read it. LaTeX is preferred to generate your report and highly recommended, but it is not mandatory. You can use any document editing software you wish, as long as the final product is in .pdf. Even if you do not use LaTeX to prepare your document, you can use LaTeX notation to mark up complicated mathematical expressions, for example, in comments in your code. Please submit all your source files, each function in a separate file. Clearly denote what each function does, which problem it belongs to, and what the inputs and the outputs are. Do not resubmit code or data provided to you or that you downloaded. Do not submit code written by others, simply reference or cite it. Identical submissions will be detected and both parties will get zero credit. In general, shorter code is better. You may include figures directly in your write-up or include them separately as .jpg, .gif, .ps or .eps files, and refer to them by filename.