10601 HW9 Write-up Name: Qifang Cai ID: qcai

Problem Statement

This homework deals with the problem of recommendation system. The idea is based on the report written by Yehuda Koren, Robert Bell and Chris Volinsky, *MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS,* Recommender-Systems-Netflix, 2009. I have been given the information about a set of users with different features and a set of movies with different genres. The goal is to learn from the training data, observing users' rating pattern and finally building a recommendation system that has good estimation of people's rating regarding different movies. Some difficulties in this homework includes cleaning the data, coming up with good models and implementing the model in an efficient manner.

Data Pre-processing and Feature Engineering

The original data all comes in .dat file format. First, I should extract all the data from files into data structures in Python. In the training data set, some rows have missing information, which needs to be filtered out. After cleaning data, I map each user's information to a vector of 1s and 0s, and each movie's information is processed in a similar fashion. The specific format is listed as below:

Matrix User

Users:	Male	Female	Age Group	Occupation	Bias
Index:	0	1	2 ~ 8	9 ~ 29	30

Matrix Movie

Movies:	Genres	Bias	Average Rating
Index:	0~17	18	19

Matrix U

Movies:	Weights of Movie Matrix
Index:	0 ~ 19

Matrix M

Movies:	Weights of User Matrix
Index:	0 ~ 30

Both Matrix User and Matrix Movie have bias terms that help increase accuracy. And the newly introduced Average Rating term in the Matrix Movie helps determine the effect of average rating score. I also have corresponding weight vectors U and M that have the size of Matrix Movie and Matrix User. These matrix U and M are the factorization matrix that I want to learn from the training data. In the prediction process, I append the User Matrix with matrix U and matrix M with Movie Matrix. The dot product of these combined matrix is the predicted rating of the user regarding a movie.

Model Implemented (including choice of parameters and how they were chosen)

The model used in this homework is a standard matrix factorization model with L2 regularization. The optimization process is done using SGD. The whole implementation is very like the whiteboard post in lecture 25. Given different tries and errors, I found the best regularization lambda value to be 0.1. In each iteration of SGD, I first randomly pick a starting

index to the training data, then go through all the training samples in order. This ensures that I cover all the possible training sets. However, given the time, I don't have a customized cross validation in place. I believe that a cross-validation mechanism can further improve this implementation's performance. After picking a single sample, I can calculate the error associated with this sample and update the entire feature weight matrix U and M using the calculation error value. U and M is updated simultaneously. And the descent step is decided based on the absolute value of error. The idea of using ALS and block coordinate descent seems promising. But I did not implement in that way. All the bias terms and feature terms are fully integrated in the user and movie matrixes. And the importance of different features is trained from observed data using SGD.

Analysis of Results (including why the metric used for analysis of this task was appropriate)

$$\sqrt{\frac{\sum_{(i,j)\in\Omega_{test}} \left(\hat{r}_{i,j} - r_{i,j}\right)^2}{|\Omega_{test}|}}$$

The task use the measure of RMSE, root mean square error. The equation is given above. This measure calculates the mean of absolute prediction differences and is appropriate to the task of recommendation system. The goal of this system is to predict the rating as closely to real world test data as possible. By minimizing the RMSE of the test data set, one is essentially improving the prediction accuracy. During training, my algorithm yields a local minimum RMSE of 0.86 on the training data. From all the submission that I have on Auto-lab, my minimum RMSE is around 0.93737. This performance is little too far from the perfect score of 0.88. There's several factors that can result in this high RMSE. First is that I might have a model that's too simplified and capture less information. Introducing only two bias terms in the matrix seems not enough. I can probably add more variables and features to capture hidden patterns. Also, I can do a better job in choosing the regularization variable Lambda. Adding cross-validation helps find the best hyper-parameters in machine learning. My implementation of SGD is not fast enough. It simply takes me too much time to train the model and converge to reasonable feature matrix. Having slow implementation limited my prediction performance as I don't have a converged model. I should start with ALS and block coordinate descent approach at the very beginning. Lastly, I should improve the regularization process, forcing more regularization on not only the matrix weight variables but also bias variables and other features. Heavy regularization helps produce simpler model that can perform better in real world prediction.

If you are in a group of two, please briefly state how you shared the workload.
I do not have a group.
Any other collaboration with students outside of your group (as detailed in the course policy).
I did not collaborate with others on this homework.
Time spent.
24 hours