CSE 6740: Computational Data Analysis
Assignment #3

Due on Thursday, November 14, 2019

Shahrokh Shahi
(GT Account: sshahi3)

Shahrokh Shahi
(sshahi3)

Q5 – Programming: Recommendation System

Part (a)

$$\frac{\partial E(U,V)}{\partial U_{u,k}} = -2 \sum_{i:(u,i)\in M} M_{u,i} V_{i,k} + 2 \sum_{(u,i)\in M} \sum_{k=1}^{r} U_{u,k} V_{i,k}$$

$$\Longrightarrow \boxed{\frac{\partial E(U,V)}{\partial U_{u,k}} = -2 \sum_{i:(u,i)\in M} \left(M_{u,i} - \sum_{k=1}^{r} U_{u,k} V_{i,k} \right) V_{i,k}}$$

And similarly,

$$\boxed{\frac{\partial E(U,V)}{\partial V_{i,k}} = -2 \sum_{i:(u,i)\in M} \left(M_{u,i} - \sum_{k=1}^{r} U_{u,k} V_{i,k} \right) U_{u,k}}$$

Therefore, the gradient descent has the following form:

$$\begin{cases} U_{u,k} \leftarrow U_{u,k} + 2\mu \left(\sum_{i:(u,i)\in M} \left(M_{u,i} - \sum_{k=1}^{r} U_{u,k} V_{i,k} \right) V_{i,k} \right) \\ V_{i,k} \leftarrow V_{i,k} + 2\mu \left(\sum_{i:(u,i)\in M} \left(M_{u,i} - \sum_{k=1}^{r} U_{u,k} V_{i,k} \right) U_{u,k} \right) \end{cases}$$

Part (b)

The new given regularized objective function:

$$E(U,V) = \sum_{(u,i)\in M}\left(M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}\right)^2 + \lambda\sum_{u,k}\left(U_{u,k}\right)^2 + \lambda\sum_{i,k}\left(V_{i,k}\right)^2$$

where λ is a hyper-parameter added to control the penalization. Now, we can write:

$$\frac{\partial E(U,V)}{\partial U_{u,k}} = -2\sum_{i:(u,i)\in M}\left(M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}\right)V_{i,k} + 2\lambda U_{u,k}$$

$$\frac{\partial E(U,V)}{\partial V_{i,k}} = -2\sum_{i:(u,i)\in M}\left(M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}\right)U_{u,k} + 2\lambda V_{i,k}$$

Accordingly, the gradient descent will have the following form:

$$\begin{cases} U_{u,k} \leftarrow U_{u,k} + 2\mu\left(\sum_{i:(u,i)\in M}\left(M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}\right)V_{i,k}\right) - 2\lambda U_{u,k} \\ V_{i,k} \leftarrow V_{i,k} + 2\mu\left(\sum_{i:(u,i)\in M}\left(M_{u,i} - \sum_{k=1}^{r} U_{u,k}V_{i,k}\right)U_{u,k}\right) - 2\lambda V_{i,k} \end{cases}$$

Part (c) - Implementation

Report:

The above-mentioned gradient descent formulation (with regulation) has been implemented in MATLAB (myRecommender.m). Choosing the learning rate (μ) is very sensitive. If we choose it a large value, it can lead to divergency and if we choose it very small, the learning procedure can be very slow, and the loop will hit the maximum iterations before meeting the proper accuracy. Therefore, at first, I set the regularizer (λ) to zero and tried to find the best value for μ. The best value has been found by checking the errors (RMSE) for a reasonable range of lowRanks (less than 10; obviously, by increasing lowRank the procedure becomes more prone to overfitting). After finding the best value for μ, I changed the values of λ in the fine-tuning step. This value will be very important when the lowRank becomes larger. In such situations, λ is an important parameter to prevent overfitting. The maximum iterations (maxIter) has been chosen in

a way to comply the execution time constraints (to be sure that it will be executed in less than 3 minutes). Of course this part mostly relies on the system specifications, but as I run the submitted code on my personal computer with usual configurations (`MacBook Pro /1.4 GHz Quad-Core Intel Core i5 / 8 GB RAM`), we can expect that it will be almost the same on the testing computer. Therefore, finally, I chose maxIter = 1000 (assuming that the evaluation testing set is almost of the same as given training and testing datasets)

Moreover, I have defined two other parameters, maxErr and maxErrRel which are maximum absolute and relative errors, respectively. I used these two values to measure the convergency rate and to define the condition to avoid unnecessary iterations by breaking the loop before reaching the maximum iterations. The following table illustrates the performance (RMSE) of my code on the training and test sets for different values of lowRank.

lowRank	RMSE		elpsed time (sec)
	training	test	
1	0.9209	0.9508	1.74
3	0.8451	0.9273	5.37
5	0.8000	0.9437	5.76
8	0.7999	0.9374	2.97
10	0.7997	0.9361	2.79
12	0.7993	0.9332	2.87
15	0.7990	0.9239	2.66

It is obvious that by increasing the lowRank, the RMSE for the training and test sets become smaller. In such situations, regularizer (λ) plays a key-role to prevent overfitting.