Programming assignment 2: implement a matrix-factorization based recommender system

In this programming assignment, you are asked to implement a matrix factorization method for rating prediction. You also need to implement some metrics to evaluate your method.

Due date:

Monday, Nov 14, 2:59pm

Here is the instruction:

1. You are provided with two files, one of which has the training data (matrix) and the other has the testing data (matrix). These two files follow the format as we discussed in the class. You need to first read in these two files and construct the CSR data representations from the files using the functions you implemented in programming assignment 1. Please note that in the input files, the indexing follows Matlab style, not C style, that is, the first index is 1, not 0.
2. Implement a matrix factorization method using alternating least squares (ALS) method. The optimization problem is

$$\min_{P,Q} \frac{1}{2} \sum_{\{u,i \in S\}} \left(r_{\{u,i\}} - p_u q_i^T\right)^2 + \frac{\lambda}{2} (|P|_F^2 + |Q|_F^2) \quad (0)$$

   a. For the rating matrix R of size m-by-n, initialize a user factor matrix P of size m-by-k and an item factor matrix Q of size n-by-k. You can use dense representation for P and Q. All the values in P and Q should be initialized either randomly or to a same number (5 points). You need to allow the users of your recommender system to specify k and λ values. That is, you have to implement a user interface (5 points) via a command line or a configuration file where the users can input the k and λ values.
   b. After you initialize P and Q, fix Q first and solve for P using either the closed form solution method or the gradient descent as we discussed in class. Then fix P and solve for Q (you need to follow this order, i.e., solve for P, solve for Q, solve for P, solve for Q, in order to get 5 points). That is, you solve

$$\min_{P} \sum_{\{u,i \in S\}} \left(r_{\{u,i\}} - p_u q_i^T\right)^2 + \frac{\lambda}{2} |P|_F^2 \quad (1)$$

   and then

$$\min_{Q} \sum_{\{u,i \in S\}} \left(r_{\{u,i\}} - p_u q_i^T\right)^2 + \frac{\lambda}{2} |Q|_F^2 \quad (2)$$

   You alternate between (1) and (2) until convergence. You need to implement a function named LS_closed or LS_gd (30 points) to solve each iteration. That is, you need to call LS_closed/LS_gd to solve for P and solve for Q in each iteration. You may need to transpose R using the transpose function you implemented in programming assignment 1. You can use the library functions to do matrix inverse.
   There are two convergence criteria. The first one is the number of iterations (a set of "solve for P, solve for Q" is considered as one iteration). If the number of iterations is larger than maxIters, then you can consider the optimization converges and you can stop the iterations. The second one is the value of the objective function (0). If its value remains stable, then you can consider the optimization converges and stop the iterations. You need to implement a user interface so that the users of your

recommender systems can specify the values of maxIters (5 points) and a threshold epsilon (5 points) such that is | f(t) – f(t-1) | / |f(t - 1)| < epsilon (f(t) is the value of the objective function (0) after the t-th iteration), the value of the objective function (0) is considered as stable. Either of the convergence criteria is satisfied (5 points), you can stop the iterations (i.e., model training).

3. After you learn P and Q, look at the testing set. If there is $r_{\{u,i\}}$ in the testing set, then do a prediction on $r_{\{u,i\}}$ using P and Q, that is, $\widehat{rr_{\{u,i\}}} = p_{\{u\}} q^T_{\{i\}}$, and calculate MSE (5 points) and RMSE (5 points). Your program needs to output MSE and RMSE into stdio together with all the parameters in the follow format (5 points):

   k = xxx, λ = yyy, maxIters = zzz, epsilon = mmm, MSE = aaa, RMSE = bbb

4. Bonus credits (20 points):
   a. If you can implement both the matrix factorization methods.
5. Submission:
   a. Source code
   b. A README.txt file containing following information, each piece in a paragraph
      i. How to compile and run your code (3 points)
      ii. A results.txt file containing the following information (3 points). If you do bonus credits, a results_bonus.txt file with the same information in the same format.
      Information:
      For k = 10, 50

      For λ = 0.01, 0.1, 1, 10

      For maxIters = 50, 100, 200

      For epsilon = 0.0001, 0.001, 0.01

      MSE value, RMSE value training time, testing time

      in the following format

      k λ maxIters epsilon MSE RMSE training_time testing_time

      Timing for model training (step 2) and testing (step 3), in ms with 2 digits after decimal point