Ethan Maddin
elm2733

CS 391L: Machine Learning

Homework 1

The purpose of this assignment was to predict missing ratings for a given set of users and movies using matrix factorization. This was done in MATLAB, and each function included in the .ipny template was reprogrammed for MATLAB. Also, the training R was factored as UM^T rather than U^TM. The code used for this assignment is included in the .zip file.

Question 1: Select λ=0 and compute U and M. What are the problems you face?

Answer: If $\lambda = 0$, then $A = X^TX + \lambda I$ is not always invertible. In other words, the matrix is not always non-singular. In terms of code output, MATLAB detects this matrix is close to singular, and gives a warning every iteration of the loop in train:

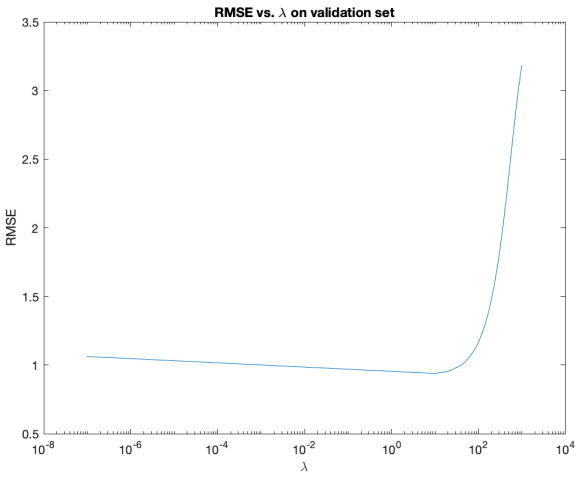
```
Warning: Matrix is close to singular or badly scaled. Results may be
inaccurate. RCOND = 6.289460e-20.
> In solve_ridge (line 6)
In update_M (line 20)
In train (line 6)
In CS_391L_HW1 (line 14)
```

Also, some of the predicted ratings matrix values are negative, which has no physical meaning in the context of this problem. Hence, this causes large RMSE values between the predicted ratings matrix and the validation matrix. The training error is low, but this doesn't have any physical meaning in the sense that the test error is much higher. It is necessary to determine an optimal lambda. Note, to solve the ridge regression problem, Gaussian Elimination or Cholesky Decomposition can be used but these are time-costly methods.

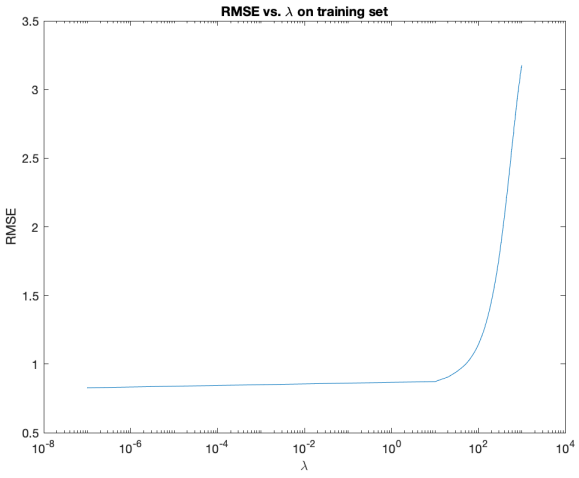
Ethan Maddin
elm2733

Question 2: Train your model on multiple values for λ and record the Root Mean Squared Error (RMSE) on both validation and train set for each run. Report the following: a) plot of λ vs RMSE on validation set and training set (both on separate plots);

Answer: RMSE vs. λ for validation set:

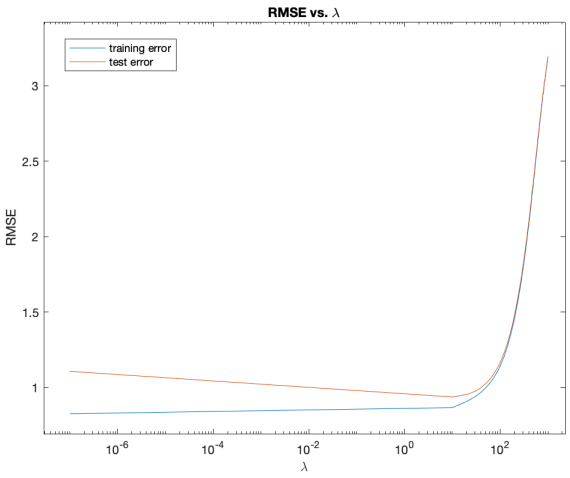


RMSE vs. λ for training set:



Combined Plot:

Ethan Maddin
elm2733



b) optimal RMSE on validation set along with the optimal λ.

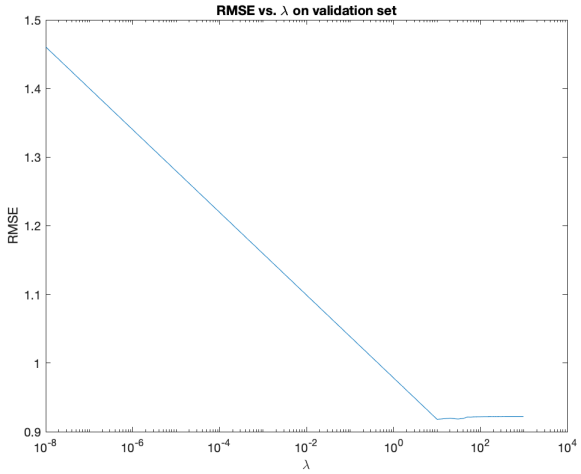
The optimal lambda and optimal RMSE were obtained using the MATLAB's min function, where min iterates through the error matrix on the validation set, and provides the smallest RMSE as well as its index. From here, the optimal λ was found using the corresponding cell index. For this particular range of λ's, the optimal RMSE was 0.9431 with a corresponding optimal λ of 10.1010. It is important to note that this was in a linspace of λ's from 10^-8 to 10^3 with a logscale on the x portion of the graph. This can also be done using MATLAB's logspace function.

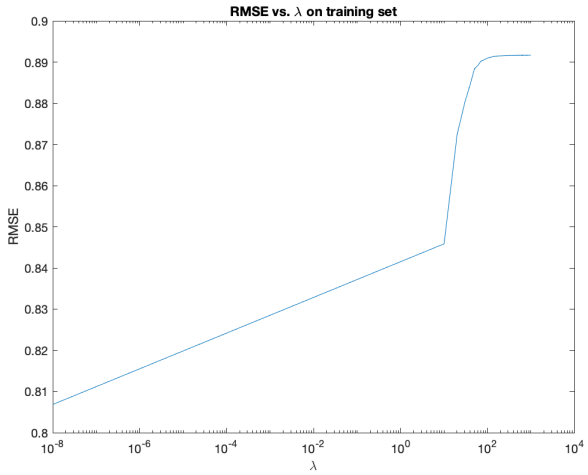
Question 3: Now let us consider preprocessing the data in order to remove some inherent bias in the data. First center the complete data using global mean. Now, center each row of the ratings matrix R to mean 0, i.e. remove user mean for each row. Similarly center each column of the resulting matrix. Perform the above operations on known ratings only. After this preprocessing, repeat question 2 above and report the optimal RMSE on validation set. Note that after factorization, you need to add the global mean and the user/movie bias for prediction. Report RMSE obtained. How does RMSE change? Explain.

Ethan Maddin
elm2733

Answer: The required operations were performed on trn_R prior to matrix factorization. It is important to note that MATLAB's sum function returns NaN if the mean of a column is 0, so this was accounted for in the post-processing when the column means were added back to the pred_R found. The new RMSE obtained was 0.9139 with a corresponding λ value of 19.1927. This is not drastically different from the optimal RMSE obtained when the data was not pre-processed through centering, but it is a lower error due to the removal of bias in the data. The optimal λ did differ by a factor of around 2. The plots shown below are using linspace of 10^-8 to 10^3 shown on a logscale similar to Problem 2.

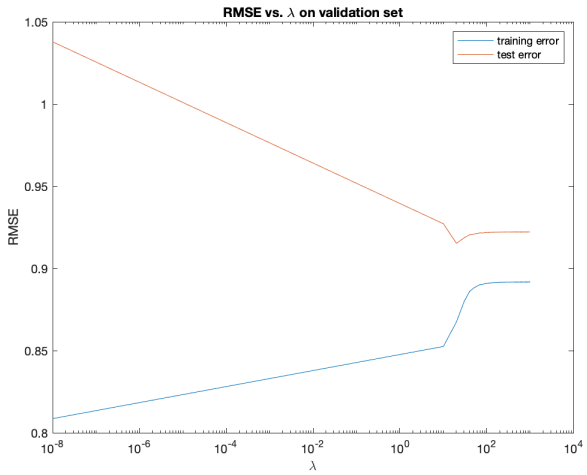
RMSE vs. λ for validation set and training set respectively:





Ethan Maddin
elm2733

Combined plot:



Question 4: What other initialization can be used? What is the problem with using the zero matrix as the initialization for U and M?

Answer: Another way to initialize this could be to find the best rank 'k' matrix from the ratings matrix and use singular value decomposition. A rank k matrix is required since U and M which are of respective dimensions m x k and n x k. However, not every rating in the training matrix is known, so SVD might not yield the best predicted ratings in the case where the ratings matrix is sparse. Something that can be toyed with is changing the initialization of U and M using difference random number functions. In the case of MATLAB, the main 3 are rand (which was used for Questions 2 and 3, and creates uniform random numbers in between 0 and 1), randn, and randi. The randn function creates random numbers on the normal distribution with mean and standard deviation of 0. When randn was used for U and M, the optimal RMSE increased to 1.5327. The randi function yield uniformly distributed integers in a range of 1 to r, where r is chosen by the user. Naturally, since ratings go from 1 to 5, r was set to 5 and then the program was run. When randi was used to initialize U and M, the optimal RMSE was 1.0817, which was

closer to the optimal RMSE found in the previous questions, but the optimal λ was multiple orders of magnitude different than the previous λ's found. For this case, the rand function yielded the best predicted ratings matrix.

One problem with initializing U and M as zeros is that the predicted ratings matrix just ends up being a matrix of zeros. In conjunction with this, RMSE never decreases and appears to stay at a constant value that is much higher than the optimal lambdas obtained in Question 2 and 3 due to the values in the predicted R remaining zero over every iteration in the training function. Diving deeper into this, when solve_ridge is called to update U and M, a matrix of 0s will always be the result no matter if one looks at user i or movie j.

Note: All the code is attached with all the functions made in the zip file. This includes the main code used to solve Questions 2 and 3.