Project 3 - Collaborative Filtering

Shubham Mittal  
Swati Arora

Anshita Mehrotra

Introduction

A collaborative filtering is a recommender system that refers to methods of predicting a user’s opinion on an entity using other users’ opinion. We predict the users with similar behavior as the target user.

Question 1- Least Square Factorization

A function is defined which takes input R matrix, latent features and maximum iterations for factorization. Matrix Factorization method generates two matrices Rmn = Umk · Vkn , such that each cell in R is generated by dot product of latent vector describing user and a latent vector describing item.

We tested for various values of latent features K= 10, 50, 100 to optimize the sizes of U and V. Following were the results obtained:

Least Squared Error for different latent features

|  |  |  |
| --- | --- | --- |
| Dimension k | Least Squared Error | Iterations |
| 10 | 2.4633e+02 | 100 |
| 50 | 1.7422e+02 | 100 |
| 100 | 1.3170e+02 | 100 |

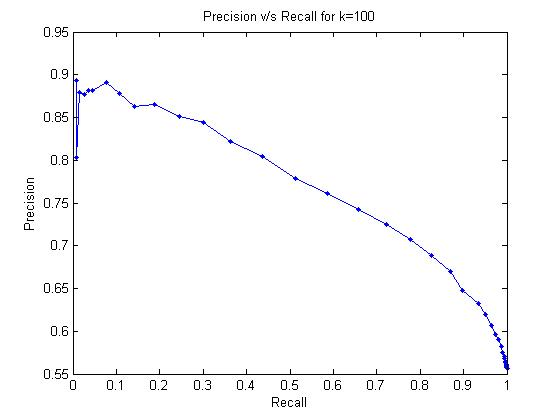
Question 2 - 10-fold Cross-Validation on Recommendation System

The recommendation system is tested using "10 - fold Cross validation". The data is split randomly as 90 percent training and 10 percent testing across 10 folds (unique test records taken for each fold).

|  |  |
| --- | --- |
| minErr | 0.8924 |
| maxErr | 0.9300 |
| Average Absolute Error | 0.9070 |

Question 3 - Recommendation Systems with Threshold Limits

Based on the prediction done in above questions, the data is now classified into two sets that is, if user likes a movie or if he does not like a movie based on the threshold values. If the predicted value is threshold that means the user likes the movie and vice versa.

Precision vs Recall Curve  


Question 4

Now in Part A of Ques. 4 we apply the same cost function as done in Ques. 1, but this time we reverse the roles of R and W matrices in the factorization step.

Least Squared Error for k = 100: 2.534044e+00

We now modify the cost function to add a regularization term lambda . We choose values of lambda to be 0.01, 0.1 and 1.0. The precision and recall was calculated for all lambda values with respect to the latent features.

Question 5

For this part, first we find predicted R matrix by supplying R as a 0-1 matrix where 1 is when a rating is available and 0 otherwise. On the other hand, the weight matrix comprised of the actual ratings that the users had given.

While performing a 10 fold cross validation, we kept a track of the predicted ratings corresponding to the known data points. The next step is to sort the ratings for every user in the descending order to get the top L movies for every user.

Hit Rate is the movies recommended by the system which are liked by the user. For this, we calculate the number of movies in L for each user that has a value above threshold. This gives us the hit rate. On the other hand, the movies recommended by the system which are not liked by the user are counted as a false alarm. Thus any rating present in L falling below the threshold which indicates the user did not like it falls under this category. We got different values of Hit Rate and False Alarm Rate by increasing the L from 1 onwards.

When the value of L hit 5, the average precision of the algorithm based on the following formula was calculated:

Precision = 𝑡𝑟𝑢𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒/ 𝑡𝑟𝑢𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒 + 𝑓𝑎𝑙𝑠𝑒 𝑝𝑜𝑠𝑖𝑡𝑖𝑣𝑒

The entire process was repeated for k = 10, 50 and 100, and the following results were obtained:

Average precision for L = 5

|  |  |
| --- | --- |
| Dimension of K | Average Precision |
| 10 | 0.531707317073171 |
| 50 | 0.549946977730648 |
| 100 | 0.556945917285261 |

As we can notice, the average precision increases with an increase in the value of k.

The Hit Rate vs False Alarm Rate curve is also an increasing one in such a manner that it approaches 1.

