Introduction to Deep learning
Assignment 1

Xintong Jiang
2898144

Wei Chen
3156486

Abstract

This report is the solution to Assignment 1. In this assignment, several recommendation algorithms were applied and their accuracy with the Root Mean Squared Error, RMSE, and the Mean Absolute Error, MAE were estimated. All of these result were tested with 5-fold cross-validation.

1 data

In this assignment the MovieLens 1M dataset fetched from http://grouplens.org/datasets/movielens/ was used. This set contains more than 1.000.000 ratings given to 3.706 movies by 6.040 users.

2 Task 1 Naive approaches

2.1 Method

Totally 5 naive approaches were applied in this task.They are:
1.average rating : global average rating of the training data was found and compared to both the training and testing data set.
2.average user rating: Average rating of each user of the training data was found and compared to both the training and testing data set. Fall back rule was used in this approach.
3.average item rating: Average rating of each movie of the training data was found and compared to both the training and testing data set. Fall back rule was used in this approach.
4.linear regression models: $pred = \alpha \cdot avg_user + \beta \dot{a}vg_movie + \gamma$ Fall back rule was used in this approach.
5.linear regression models: $pred = \alpha \cdot avg_user + \beta \dot{a}vg_movie$ Fall back rule was used in this approach.

The predictions were improved by rounding values bigger than 5 to 5 and smaller than 1 to 1. Fall back value was the global average of the training set.

2.2 running time

The actual running was get by doing ten replicates and taking the average.

	cpu-time	memory	actual run time
average rating	O(N)	O(1)	1.18s
average user rating	O(NlogN)	O(U)	2.05s
average item rating	O(NlogN)	O(M)	2.24s
linear regression with intercept	$O(N^3)$	$O(1)$	6.13s
linear regression without intercept	$O(N^3)$	$O(1)$	5.98s

Table 1: time and space

.

N is the total number of ratings, U is number of unique users and M is number of unique movies.

2.3 result

In this task random seed is 2898144.

	RMSE Training	MAE Training
average rating	1.117	0.934
average user rating	1.028	0.823
average item rating	0.974	0.778
linear regression with intercept	0.915	0.725
linear regression without intercept	0.947	0.759

Table 2: Training Error

.

	RMSE Testing	MAE Testing
average rating	1.117	0.934
average user rating	1.035	0.829
average item rating	0.979	0.782
linear regression with intercept	0.924	0.732
linear regression without intercept	0.952	0.763

Table 3: Testing Error

We can find that the accuracy of the model from high to low is: linear regression, average item rating, average user rating, global average rating. And the model linear regression with intercept is better than the linear regression without intercept. This is because the former model has 1 more parameter than the latter model and can model the bias of the data, it fits better in this case.

3 Task 2 UV-Decomposition

The blank entries in our rating matrix **M** can also be estimated by conjecturing rating matrix to a product of two long, thin matrices **U**,**V** and minimize the RMSE.
A certain value of **U** affects only the elements in it's row of the product **P** = **U** **V**. If we replaced U_rs by a variable x, P_rj can be computed as follow:

$$P_{rj} = \sum_{k=1}^{d} U_{rk} v_{kj} = \sum_{k \neq s} U_{rk} v_{kj} + x V_{sj}$$

$\sum_{k \neq s}$ is the sum for k = 1, 2, . . . , d, except for k = s.
And the error between matrix **M** and **U** can be computed as:

$$(M_{rj} - P_{rj})^2 = (M_{rj} - \sum_{k \neq s} U_{rk} v_{kj} - x V_{sj})$$

If we want to minimize this error, we need to take the derivative of it with respected to x, and set it equal to 0.

$$x = \frac{\sum_j V_{sj}(M_{rj} - sum_{k \neq s} U_{rk} V_{kj})}{\sum_j V_{sj}^2}$$

Follow the same procure, if we want to find the value of y from matrix **V** that minimizes the RMSE.

$$y = \frac{\sum_i U_{ir}(M_{is} - sum_{k \neq r} U_{ik} V_{ks})}{\sum_i U_{ir}^2}$$

3.1 method

1. Preprocessing of the matrix:
Every non-blank element of matrix M: m_{ij} was subtracted the average rating of user i. Then, the result was subtracted the average rating of movie j. If the average rating of user i or the average rating of movie j is nan in the training set, the fall-back value was set to the global average of the training set. After the predict values were obtained, the average rating of user i and the average rating of movie j that had been subtracted were plus back.

2. Initializing **U** and **V**:
The starting point for **U** and **V** is a normally distributed value with mean 0 and a standard deviation that is same to the standard deviation of our processed training set.

3. Ordering the optimization of the elements of **U** and **V**:
A permutation of the elements was chosen every round. Each element was optimize following this order.

4. Ending the attempt at optimization:
The iteration ends, if the improvement of the RMSE and MAE does not change much. Here the epoch is 50.
To avoiding over-fitting, a component was only moving 0.3 of the way from its current value toward its optimized value.

3.2 running time and space

The complexity in time is O(UVd) without considering iterations,and complexity in memory is O((U+V)*d).Where U is and V is the long side of matrix **U** and **V**. d is the short side. The actual running time when d=2 an epoch=50 is 1441s.

3.3 result

After 50 iterations, we can compare result of different value of d, where d is the short side of the matrix **U** and **V**.

	RMSE Training	MAE Training
d=2	0.863	0.678
d=3	0.844	0.662
d=4	0.829	0.649

Table 4: Training Error

.

	RMSE Testing	MAE Testing
d=2	0.898	0.703
d=3	0.891	0.696
d=4	0.891	0.694

Table 5: Testing Error

d=2 has a better performance than d=3, and the performance of d=3 and d=4 are similar. Since d=3 is faster,we choose d=3. The final RMSE for the testing set is 0.891 and the MAE is 0.696.

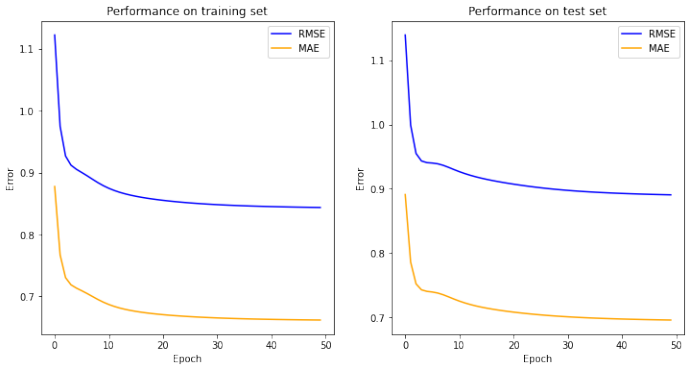

Figure 1: Performance of UV decomposition

4 Task 3: Matrix Factorization

4.1 Algorithm

This algorithm described in the *On the Gravity Recommendation System* paper is implemented and we experiment with the required hyper-parameters:

- num_factors=10, num_iter=75
- regularization=0.05, learning_rate=0.005

This algorithm can be roughly summarized as follows:

- For initial $I \times J$ matrix **R** where I and J denote the number of the users and movies, while values represent users rating for movies, we randomly initialize two matrices **U** $(I \times K)$ and **M** $(K \times J)$ to approximate the matrix **R** where K denotes the number of features or factors. The random seed here for generating **U** and **M** is 33.
- Let r_{ij}, u_{ik} and m_{kj} represent values or weights in matrix **R**, **U** and **M** respectively. The prediction of the matrix **R** is the product of **U** and **M**.

  $$pred = \sum_{k=1}^{K} u_{ik} m_{kj}$$

  Represent error by subtracting $pred$ from r_{ij}.

  $$e_{ij} = r_{ij} - u_{ik} m_{kj}$$

  where u_{ik} and m_{kj} are row vectors and column vectors.
- Due to **R** is a sparse matrix, we simply iterate over all known values of it and then update the corresponding row vector in **U** and column vector in **M** to find local minimum using gradient descent.

  $$u_{ik} = u_{ik} + \eta * (2e_{ij} * m_{kj} - \lambda * u_{ik})$$

  $$m_{kj} = m_{kj} + \eta * (2e_{ij} * u_{ik} - \lambda * m_{kj})$$

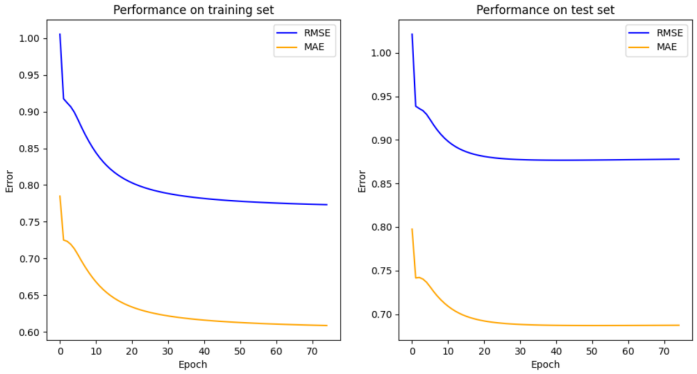

Figure 2: Performance of Matrix factorization

- Compute RMSE and MAE to observe training process.

Finally, by $\mathbf{U}\cdot\mathbf{M}$, we obtain the prediction and squeeze each element to [1, 5] after 75 iterations. For this algorithm (except for cross validation) the complexity in time is *O(IJK)* without considering iterations, and complexity in memory is *O(IK+JK)*.

4.2 Cross Validation

We use dictionary to store the data set of ratings, where keys are the identity of each user, and corresponding values are the index of each movie. Next, Data set is shuffled (seed = 88) and for all users, the idea of 5-fold cross validation is applied to respectively split their index of movies into five pieces. Then combine all users with their index of movies, that is, each combination of 4 out of 5 pieces and each combination of 1 out of 5 pieces, as training set and test set (repeat five times). Next we launch five processes to create five models on these five parts (folds).

4.3 Performance

After given iterations, we save parameters ($\mathbf{U}$ and $\mathbf{M}$), as well as training results and testing results to local disk to ease the way for us to observe the performance of this algorithm and check whether the selected hyperparameters are appropriate.
Furthermore, for the algorithm *Matrix factorization with regularization using gradient descent*, it takes about 30 minutes to run 75 iterations. The running time of this algorithm is computed by Jupyter. The average RMSE and MAE of these five models on test set is *0.878* and *0.687* respectively, which is similary to the results reported in MyMedialite (See Fig. 2).