Implementations of Recommendation System with

different algorithms on MovieLens Dataset

Minyang Tie

Computer and Data Sciences

Case Western Reserve University

Cleveland, Ohio, USA

mxt497@case.edu

ABSTRACT

In this document’s first part, we implement five algorithms to construct a recommendation system. In each algorithm, we first show the performance of this algorithm on specific hyperparameters. Then we go to a model selection procured. We plot the performance on test dataset of different hyperparameters then we figure out which is best. In the second part, we compared the different the performance of different algorithms and make some simple but with insight conclusions then proposed some advice may be useful to enhance the performance.

1. Algorithm Implementations

1.1 Random

This algorithm predicting a random rating based on the distribution of the training set, which is assumed to be normal. There prediction is generated from a normal distribution where andare estimated from the training data using Maximum Likelihood Estimation

|  |  |  |  |
| --- | --- | --- | --- |
|  | RMSE | MAE | Test Time |
| Fold1 | 1.5152 | 1.2155 | 0.15 |
| Fold2 | 1.5154 | 1.2142 | 0.12 |
| Fold3 | 1.5276 | 1.2289 | 0.14 |
| Fold4 | 1.5208 | 1.2227 | 0.12 |
| Fold5 | 1.5124 | 1.2115 | 0.14 |
| Mean | 1.5183 | 1.2186 | 0.13 |
| std | 0.0054 | 0.0064 | 0.01 |

**table 1.1 the 5 fold cross validation performance of random algorithm**

1.2 KNN

This algorithm prediction a rating of one specific user and movie based on the neighboring movies’ rating which have the similar movie tags. The algorithm can be formulated as follows:

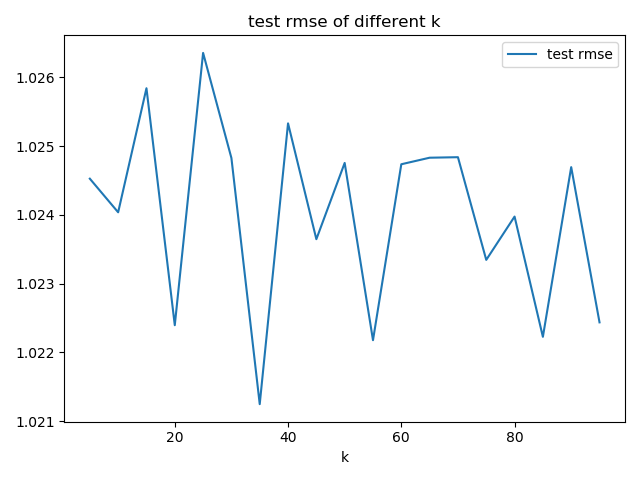


Following table show the performance of this algorithm where ***k*** = 90

|  |  |  |  |
| --- | --- | --- | --- |
|  | RMSE | MAE | Test Time |
| Fold1 | 1.0206 | 0.8044 | 1.96 |
| Fold2 | 1.0341 | 0.8118 | 1.96 |
| Fold3 | 1.0275 | 0.8066 | 1.87 |
| Fold4 | 1.0274 | 0.8084 | 1.82 |
| Fold5 | 1.0213 | 0.8013 | 1.78 |
| Mean | 1.0262 | 0.8065 | 1.88 |
| std | 0.0049 | 0.0036 | 0.07 |

**table 1.2 the 5 fold cross validation performance of knn algorithm where k = 90**

As k is a hyperparameter of this algorithm, we also explored which k is best. Following figure shows the performance of the mean rmse of 5 fold cross validation of different k.



**figure 1.2 the performance of knn of different k**

we can see that the performance doesn’t change a lot. So maybe k = 35 is the best option for MovieLen dataset.

1.3 SVD

This algorithm prediction a rating can be formulated as following formulations :



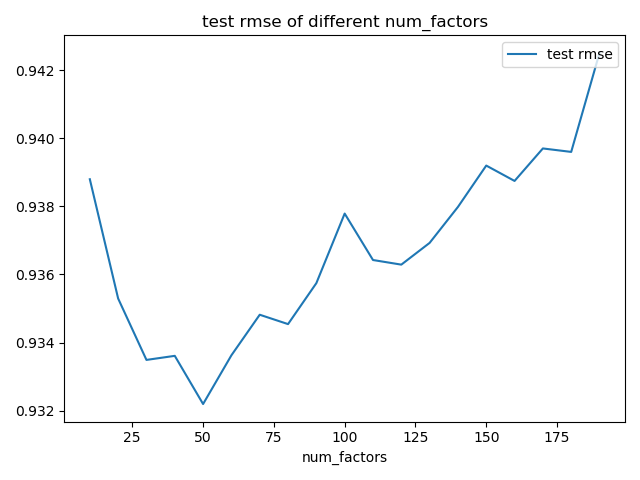
If user is unknown, then the bias  and the factors are assumed to be zero. The same applies for item I with and 

Following table show the performance of this algorithm where hyperparameter ***num\_factors =*** 4

|  |  |  |  |
| --- | --- | --- | --- |
|  | RMSE | MAE | Test Time |
| Fold1 | 0.9386 | 0.7413 | 0.20 |
| Fold2 | 0.9310 | 0.7358 | 0.16 |
| Fold3 | 0.9374 | 0.7395 | 0.23 |
| Fold4 | 0.9383 | 0.7399 | 0.14 |
| Fold5 | 0.9377 | 0.7404 | 0.13 |
| Mean | 0.9366 | 0.7394 | 0.17 |
| std | 0.0028 | 0.0019 | 0.04 |

**table 1.3 the 5 fold cross validation performance of svd algorithm where num\_factors = 40**

As ***num\_factors*** is a hyperparameter of this algorithm, we also explored which ***num\_factors***  is best. Following figure shows the performance of the mean rmse of 5 fold cross validation of different ***num\_factors***.



**figure 1.3 the performance of svd of different num\_factors**

we can see that num\_factors = 50 performs best. And the test rmse increased with the increment of num\_factos.

1.4 Matrix Factorization

This algorithm assume that every user and movie both have some features of the same dimension. If we use denote the user feature matrix, denote the movie feature matrix. (note that have the dimension of num\_user num\_feature,  have the dimension of num\_movie num\_feature. Then the rating of user to the movie  can be calculated as :

**

The rating matrix  can be calculated as :

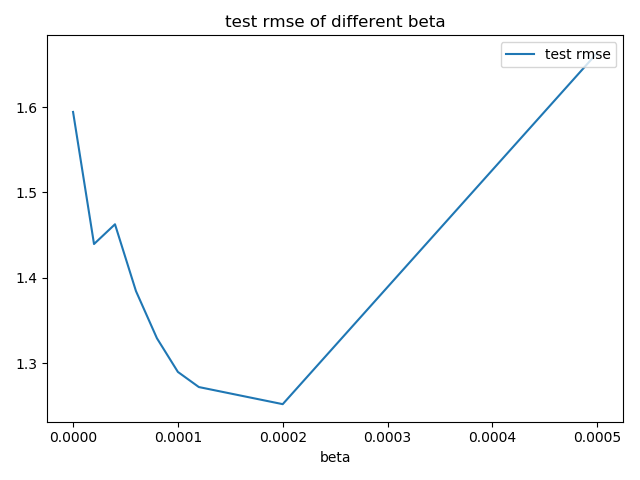
**

This algorithm tries to calculate the matrix  and  add some regularization to  and . The regularization coeffect is denoted as 

Following table show the performance of this algorithm where hyperparameter ***=*** 2e-3

|  |  |  |  |
| --- | --- | --- | --- |
|  | RMSE | MAE | Test Time |
| Fold1 | 1.2497 | 1.0027 | 0.06 |
| Fold2 | 1.2547 | 1.0068 | 0.08 |
| Fold3 | 1.2488 | 1.0054 | 0.14 |
| Fold4 | 1.2539 | 1.0095 | 0.13 |
| Fold5 | 1.2530 | 1.0069 | 0.10 |
| Mean | 1.2520 | 1.0063 | 0.10 |
| std | 0.0023 | 0.0022 | 0.02 |

As  is a hyperparameter of this algorithm, we also explored which is best. Following figure shows the performance of the mean rmse of 5 fold cross validation of different 

**figure 1.4 the 5 fold cross validation performance of matrix factorization algorithm with different beta**

we can see that  = 2e-3 performs best. And the test rmse increased with the increment of .

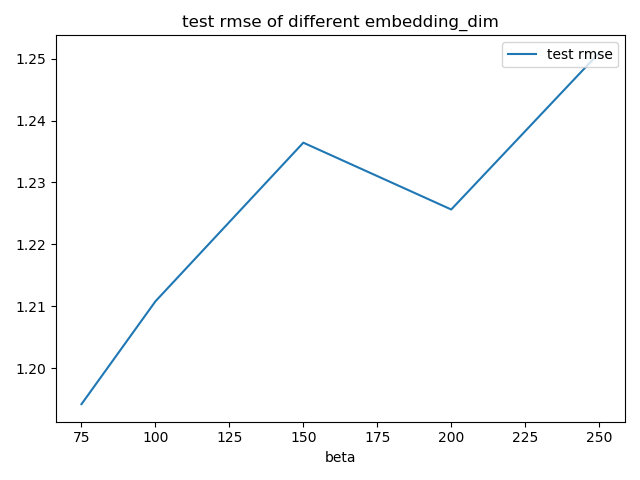
1.5 Deep Neural Network

This algorithm emended the user and movie into a high dimension space. And use the embedding of user and movie as the input retire. Then use a neural network to do the regression

Following table show the performance of this algorithm where hyperparameter ***embedding\_dim =*** 100

|  |  |  |  |
| --- | --- | --- | --- |
|  | RMSE | MAE | Test Time |
| Fold1 | 1.2527 | 1.0081 | 0.33 |
| Fold2 | 1.1820 | 0.9476 | 0.36 |
| Fold3 | 1.1789 | 0.9456 | 0.36 |
| Fold4 | 1.1756 | 0.9414 | 0.38 |
| Fold5 | 1.1814 | 0.9530 | 0.35 |
| Mean | 1.1941 | 0.9591 | 0.36 |
| std | 0.0293 | 0.0247 | 0.15 |

As ***embedding\_dim*** is a hyperparameter of this algorithm, we also explored which ***embedding\_dim***  is best. Following figure shows the performance of the mean rmse of 5 fold cross validation of different ***embedding\_dim***.



**figure 1.5 the 5 fold cross validation performance of dnn algorithm with different beta**

we can see that ***embedding\_dim***  = 75 performs best. And the test rmse increased with the increment of ***embedding\_dim.***

1. Comparison, conclusion and future work

Following table shows the rmse compared with each pair of algorithm:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Random | Knn | Svd | MF | Dnn | Mean |
| Random | 0 | 1.1592 | 1.2047 | 1.4477 | 1.2603 | 1.0144 |
| Knn | 1.1592 | 0 | 0.1056 | 1.2007 | 0.9488 | 0.7429 |
| Svd | 1.2047 | 0.4056 | 0 | 1.2265 | 0.9922 | 0.7658 |
| MF | 1.4477 | 1.2007 | 1.2265 | 0 | 1.0855 | 0.9921 |
| DNN | 1.2603 | 0.9488 | 0.9922 | 1.0855 | 0 | 0.8574 |

We can see that knn and svd have the smallest mean rmse compared with other algorithms. and the random algorithm has the highest rmse to other algorithms. That means the random algorithm predict rating far away from other algorithm. According to the statement we have mentioned below ,we can see that svd performs best. And knn performs similarly. The matrix factorization has the smallest test time. So if we do not want to a too precise model, we can use the matrix factorization algorithm to predict rating in reality world. The not good performance of dnn indicates that the now neural network structure may need to be improved.

REFERENCES

[1] FNM Surname (2018). Article Title. Journal Title, 10(3), 1–10.

[2] F.N.M Surname, Article Title, https://www.acm.org/publications/proceedings-template.

[3] F.N.M Surname and F.N.M Surname, 2018 Article Title, The title of book two (2nd. ed.). Publisher Name, City, State, Country.

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00