Advamced Machine Learning
Homework 2

Kolmakova Elizaveta
Innopolis University
Kazan, Russia
e.kolmakova@innopolis.university

***Index Terms*—Recommendation system, matrix factorization, sparse matrix, SGD, MSE, deep networks, comparison**

I. Introduction and Motivation

This is a comprehensive report-comparison of two approaches to recommendation system by means of collaborative filtering. Collaborative filtering is one of the most interesting area of researches, which could provide with relatively robustness. The two approaches are statistical-related and deep neural networks related. In the first one we use logic of gradient descent algorithm to obtain two separate matrix to calculate predicted rating value from other users. In the second approach we rely on machine learning and simply use neural networks.

The work is constructed from the following sections: section 2 briefly describes theoretical aspects and architecture of every model, section 3 introduce experiment settings, section 4 shows results from comparison of approaches and section 5 represent conclusions with discussion.

1) Section 1. Introduction and Motivation.
2) Section 2. Theoretical aspects and architectures.
3) Section 3. Experiments settings.
4) Section 4. Results.
5) Section 5. Conclusion and discussion.

II. Theoretical aspects and architectures.

A. Matrix Factorization

The theory is presented in task description. Shortly, initial matrix as an input goes into stochastic gradient descent algorithm, tracking losses, with regularization. Then based on computed latent matrices recommendation rating will be build. Before the main training there is grid-search logic algorithm for identification of the best parameters of this model: latent dimension, learning rate, regularization parameter, amount of iterations.

B. Deep Learning Based Method

This approach is based on neural network with three sequence hidden layers. The sequence is next pipeline: linear layer - $ReLU activation - dropout$.

III. Experiments settings

The task is to build a recommendation system based on other users' ratings.

The initial data is non-sparce table with users' ids, movies' ids and ratings. This data was extracted and turned into sparse matrix. For sake of computation efficiency then it was turned into non-sparse matrix.

In order to conduct a validation there was a split on training and test sets. The training set was used for model weights construction, while data from test set was tested in constructed recommendation system.

A. Matrix Factorization

In order to obtain the best parameters which would provide with the highest performance measured by mean square error algorithm, the simple version of GridSearch was applied. GridSearch is simple algorithm of brute force of all parameters. There were 3 parameter's groups: latent dimension, learning rate, regularization parameter

There is a presentation of different parameters group and combinations with the best accuracy and the best convergence rate.

1) Learning rate: 0.0005, 0.0001, 0.001, 0.01, 0.1, 0.2
2) Latent dimension: 50, 16, 4, 2, 1
3) Regularization parameter: 0, 0.0001, 0.001, 0.01, 0.1, 0.2, 0.8

The tables I, II, III, represent most stable combinations with initial and final mean square error values:

Based on the result presented in tables I, II, III, the following conclusions were obtained:

1) Too high value of learning rate produce instability of gradient descent algorithm. Only high value of regularization parameter could add a little stability. Therefore, value 0.01 were chosen as preferable for this task.
2) According to results, regularization parameter does not impact much on performance. Therefore the value might be chosen 0, but it does not imply regularization is not needed there. Value 0.0001 were chosen as preferable for this task.
3) Choice of latent dimension is quite trade-off: the higher value, the lower MSE but higher time calculation. Since model can be trained only once, time calculation efficiency is not that important as value of losses. Therefore, value 50 were chosen as preferable for this task.

Number	Initial MSE	Final MSE	Learning rate	Latent dimension	Regularization parameter
1	1.0288	0.9284	0.0001	50	0
2	1.0288	0.9284	0.0001	50	0.001
3	1.0288	0.9285	0.0001	50	0.01
4	1.0289	0.9327	0.0001	50	0.2
5	0.9593	0.8542	0.001	50	0
6	0.9593	0.8542	0.001	50	0.001
7	0.9594	0.8545	0.001	50	0.01
8	0.9621	0.8625	0.001	50	0.2
9	0.8712	0.5490	0.01	50	0
10	0.8713	0.5497	0.01	50	0.001
11	0.8714	0.5767	0.01	50	0.01
12	0.8785	0.8503	0.01	50	0.2

TABLE I: Parameters according to GridSearch for latent dimension 50

Number	Initial MSE	Final MSE	Learning rate	Latent dimension	Regularization parameter
2.1	1.0289	0.9283	0.0001	16	0
2.2	1.0289	0.9283	0.0001	16	0.001
2.3	1.0289	0.9285	0.0001	16	0.01
2.4	1.0290	0.9327	0.0001	16	0.2
2.5	0.9593	0.8528	0.001	16	0
2.6	0.9593	0.8527	0.001	16	0.001
2.7	0.9595	0.8533	0.001	16	0.01
2.8	0.9622	0.8623	0.001	16	0.2
2.9	0.8707	0.6767	0.01	16	0
2.10	0.8708	0.6761	0.01	16	0.001
2.11	0.8710	0.6809	0.01	16	0.01
2.12	0.8779	0.8503	0.01	16	0.2

TABLE II: Parameters according to GridSearch for latent dimension 16

Number	Initial MSE	Final MSE	Learning rate	Latent dimension	Regularization parameter
3.1	1.0366	0.9347	0.0001	4	0
3.2	1.0363	0.9344	0.0001	4	0.001
3.3	1.0361	0.9344	0.0001	4	0.01
3.4	1.0355	0.9357	0.0001	4	0.2
3.5	0.9659	0.8535	0.001	4	0
3.6	0.9659	0.8533	0.001	4	0.001
3.7	0.9663	0.8535	0.001	4	0.01
3.8	0.9671	0.8617	0.001	4	0.2
3.9	0.8728	0.7718	0.01	4	0
3.10	0.8731	0.7678	0.01	4	0.001
3.11	0.8730	0.7711	0.01	4	0.01
3.12	0.8779	0.8499	0.01	4	0.2

TABLE III: Parameters according to GridSearch for latent dimension 4

Epoch	MF	Deep learning method
1	0.8713755757550058	0.41606050729751587
2	0.8489076373514106	0.3648034930229187
3	0.8342316135484011	0.4492085576057434
4	0.8183214451102465	0.3463340103626251
5	0.797695139426937	0.340408056974411
6	0.7486140922164758	0.3296390473842621
7	0.6975245361599808	0.3273341655731201
8	0.6526459544760697	0.34021079540252686
9	0.6156433818977302	0.37208425998687744
10	0.5858640908240429	0.36402061581611633
11	0.5502362979754198	-
Computation time	585 sec	4745 sec
Average time per top 20 recommendation computing	0.07566 sec	0.083882 sec

TABLE IV: Approaches comparison

B. Deep learning based method

The architecture of deep neural network for this task is simple: initially there are embedding layers for users and movies, then there is sequence of three sequences of linear layers with dropout and relu activation. As an output there is simple layer with sigmoid activation. The output value will be scaled from 1 to 4.

After training this architecture of neural networks the weights were saved and then reloaded to apply it on test dataset in order to understand the performance of proposed model. The performance of proposed models are described in the following section.

IV. Results

The comparison of two models is based on time of computation and loss values. The following table IV represent values for 10 epoches/iterations for two different approaches. Based on the results the following was concluded:

1) Deep learning model more robust in recommendation calculations, but more time-consuming at the moment of model training.
2) Deep learning model more reliable in terms of accuracy of prediction since the values of losses are lower in comparison with matrix factorization model.
3) Deep learning model is more computationally consuming therefore the model is more heavy.
4) Matrix factorization model is easier to compute and store. But the price of such simplicity is higher losses in comparison with deep learning model.

V. Conclusions

To sum up, the following conclusion might be obtained:

1) When there is no need in as precise values of recommendation as it possible and we track computational efficient, then simple models like matrix factorization are preferable. It especially good in terms of small dataset since its efficiency and simplicity. But the bigger dataset, the longer and more resource-consuming the calculation

will be. As well it can be applied in the systems where recommendation aspect is not prevailing, but rather an additional tool.

2) In the systems where the accuracy of prediction is important it is preferable to use deep learning approach. Such approach is harder to compute, but the prediction is much reliable. Big advantage of the approach is its intellectual aspect - it is capable to give recommendation to new user without model recomputation when new user is added.