Recommender system with biased matrix factorization

Markus Unkel[1]

[1] Friedrich-Schiller-University Jena, Master Computational and Data Science

This report is part of the lab course of "Statistical Learning Theory" in summer 2018.

Motivation. We consider the problem of predicting user dependent numerical item-ratings. Here the rating space is described by natural numbers between zero and four. In general the rating parameter is ordinal, such that the rating of an item grows with its rating number. In this report we try to solve the described problem by an assumption of splitting the user item rating-matrix into user-/item-dependent matrices via a biased matrix factorization performed by stochastic gradient descent.

Predictor. We assume the following structure of generating ratings made by an user given to an item as follows(1):

$$p : U \times I \to R \ , \ (u,i) \mapsto \bar{r} + b_U(u) + b_I(i) + x(u)^\top y(i) \quad [1]$$

with the user/item/rating space $U/I/R$, the mean rating $\bar{r}$ of given sample set D, the biased rating parameters $b_{U/I} \in \mathbb{R}^{|U/I|}$ and rating vectors $x, y \in \mathbb{R}^f$. This prediction allows a conditional decision how to rate when a qualifying user/item or both not exist, e.g. a non existing item leads to $r(u) = \bar{r} + b_U(u)$. This prediction assumption corresponds to the matrix factorization

$$\tilde{R}^{|U|\times|I|} = X^\top Y \quad [2]$$

where

$$\tilde{R}^{|U|\times|I|} = R^{|U|\times|I|} - \bar{r}^{|U|\times|I|} - b_U \cdot 1^\top - 1 \cdot b_I^\top, \quad [3]$$

with the rating matrix R and the column wise user/item vectors in $X/Y \in R^{f\times|U/I|}$. The sample set D describes in reality a sparse matrix R. The user/item parameters $b_{U/I}$ describe the trend rating for an user/item over all items/users.

Minimization Problem. To perform the matrix factorization, equation 2 can be formulated into a non convex minimization problem spaned by the data sample points $d = (u,i,r) \in D$

$$\begin{aligned} &\min_{b_U,b_I,x,y} \|\tilde{R}^{|U|\times|I|} - X^\top Y\|_D^2 \\ &= \min_{b_U,b_I,x,y} L = \sum_{d\in D} L_d = \sum_{(u,i,r)\in D} \left(r(u,i) - \hat{p}(u,i)\right)^2 \\ &\quad + c\left(|b_U(u)| + |b_I(i)| + \|x(u)\|^2 + \|y(i)\|^2\right) \end{aligned} \quad [4]$$

with the ridge parameter $c > 0$ for having a smaller finited condition number (prevents overfitting). The local extrema (hopefully local/global minimum) condition is fulfilled by the equation

$$\nabla^{b_U,b_I,x,y} L = 0, \quad [5]$$

since the problem is non convex.

Stochastic Gradient Descent. Instead of performing the gradient as the sum over the large number of data points ($|D| = \mathcal{O}(10^5)$), here we use the hopefully faster convergating method of stochastic gradient descent where the gradient ∇L is iterative approximated by a single point gradient ∇L_d. With a learning rate $\eta > 0$ every new gradient step updates the parameters (b_U, b_I, x, y) in a following way:

$$(b_U, b_I, x, y)_{t+1} = (b_U, b_I, x, y)_t - \eta \nabla L_{d(t)}(b_U, b_I, x, y)_t \quad [6]$$

with $t \in [|D|]$ and the random data sequence $d(t) \in D$. The data point dependent gradient update can be written as

$$\begin{aligned} \hat{r}(u,i) &= p(u,i)_{(b_U,b_I,x,y)} \\ bias(u,i) &= r(u,i) - \hat{r}(u,i) \\ b_U(u) &\mathrel{+}= \eta(bias(u,i) - c \cdot b_U(u)) \\ b_I(i) &\mathrel{+}= \eta(bias(u,i) - c \cdot b_I(i)) \\ x(u) &\mathrel{+}= \eta(bias(u,i) \cdot y(i) - c \cdot x(u)) \\ y(i) &\mathrel{+}= \eta(bias(u,i) \cdot x(u) - c \cdot y(i)) \end{aligned} \quad [7]$$

Monte Carlo Cross-Validation Grid Search. Next to the minimization problem the outer parameters f, η, c have to be chosen from a defined set of possible parameters. Therefore we use a cross-validation train-/test scenario where the minimum mean of all test root mean squared errors (RMSE) defines the best grid point. The the train/test data samples are generated in every run of monte carlo cross-validation nonrecurrent and randomly. For having a "good" test scenerio the number of total monte carlo runs should be in magnitude of hundreds or thousands.

Analysis

Data Set. We consider the following data set with $|U| = 1666$, $|I| = 1194$, $|D| = 434641$ and its distributions shown in figures 1- 4.

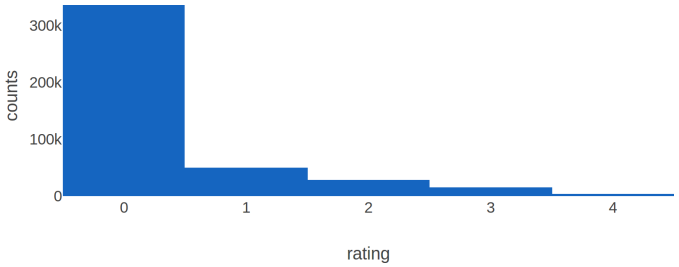

Abb. 1. Rating distribution of data set

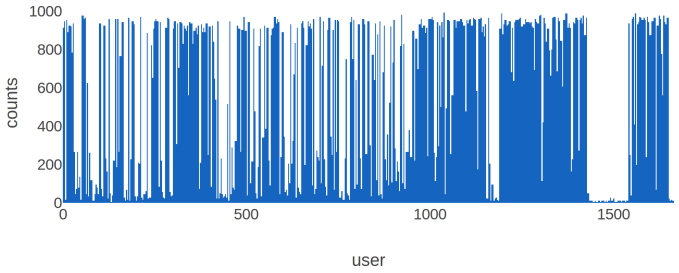

Abb. 2. User data points distribution with transformed user id

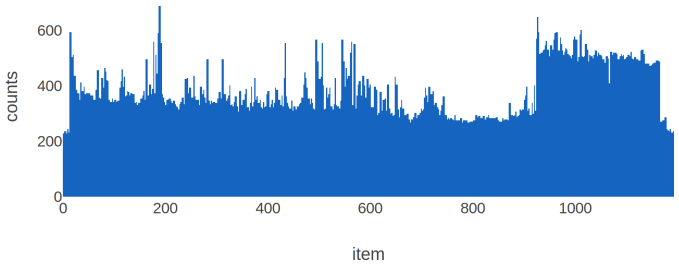

Abb. 3. Item data points distribution with transformed item id

The rating distribution shows that the algorithm will be concentrated on predicting many zeros, because the tradeoff to make good predictions for the rating four is not very helpful for minimizing L.

Grid Search. As a first hyper parameter grid we search for points within $f \in \{50, 100, 200\}$ and $c, \eta \in \{0.01, 0.02, 0.03, 0.04\}$. The program runs on an instance with 2 x Intel Xeon 6140 18 Core 2,3 Ghz and 72 GB RAM through this grid with a maximal iteration number 50, 72 monte carlo runs and a train ratio of 0.8. The total computation time is around 500 minutes. The result for the three different numbers of f are shown in figure 5. In the lower right corner of the f 50 grid search the learning rate leads to overflow problems. Here the best grid point is chosen with a test RMSE of 0.466 and train

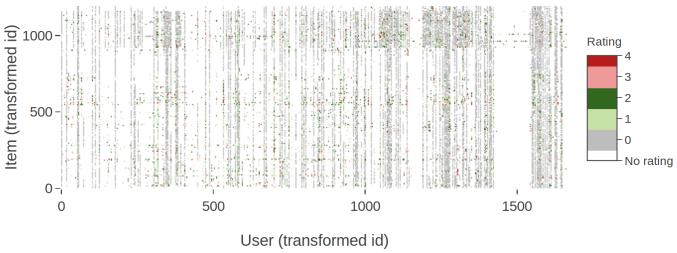

Abb. 4. Heatmap Data Rating Matrix

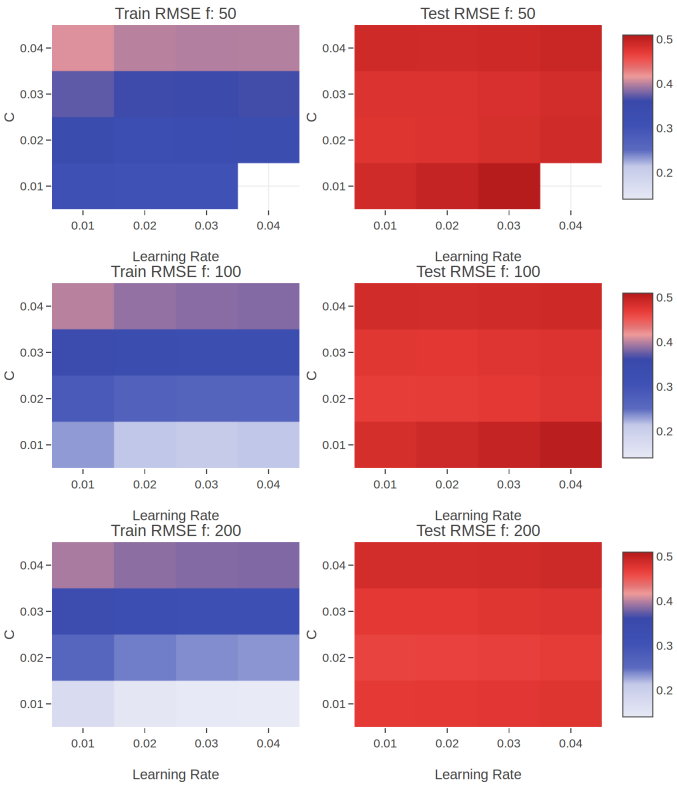

Abb. 5. Grid search with 72 monte carlo runs and max. iter = 50

RMSE of 0.262 at f = 200, c = 0.02 and $\eta = 0.01$. As a second more accurate and better located grid search we use now 100 maximal iterations in the grid $f \in \{100, 150, 200\}$, $c \in \{0.001, 0.005, 0.01, 0.015\}$ and $\eta \in \{0.005, 0.01, 0.015, 0.02\}$. The results are not very differing from the first grid search, see in figure 6. In

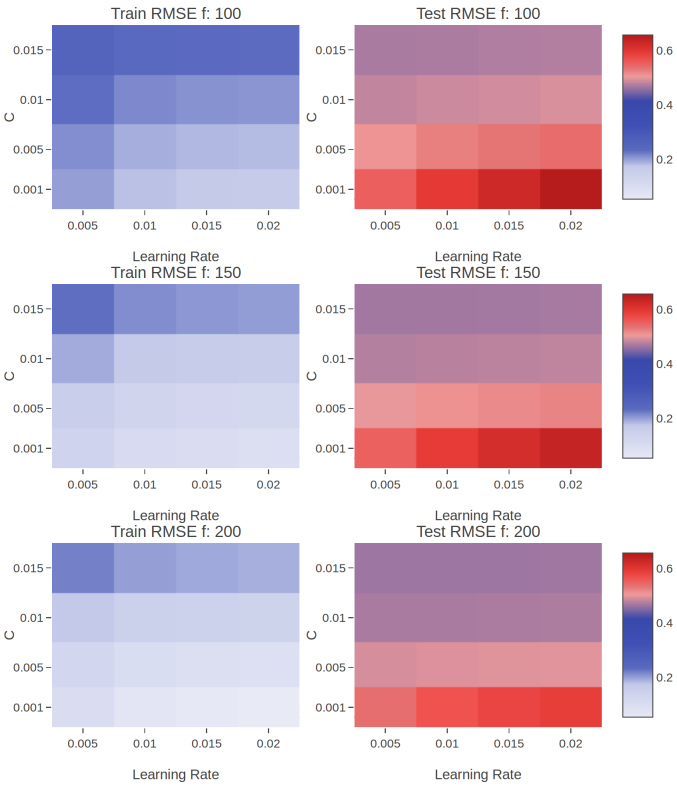

Abb. 6. Grid search with 72 monte carlo runs and max. iter = 100

general we can see that c has an elementary impact on the model overfitting, such in the way that the train RMSE is going down and the test RMSE up. In the case of $f = 200$ it is essentially that $c > 0$ because now we have more parameters than data points.

Build Model. The program chooses the grid point at f = 200, c = 0.015 and $\eta = 0.005$ and trained the final model withhelp of the total data set with a maximal iteration number of 200 (see RMSE vs iteration in figure 7). The qualifying data set contains users that not exist in the train data set (unknown users: 178, 307, 967, 1502, 1680, 3425, 4304, 4310). The RMSE of the qualifying data set is 0.485. This value is around the expected test RMSE of grid search. Here the stochastic gradient descent is only run by one thread instead of the grid search phase, where all threads worked in parallel for its own monte carlo run. A way to perform the minimization problem in parallel is to transform in a alternating way the minimizing problem to a convex problem by holding one side of the parameters constant. Then we have a traditional linear regression problem that can be solved in parallel for all users/items. This method was used at first but the convergence was reasoned by this greedy algorithm in a global problem very poor.

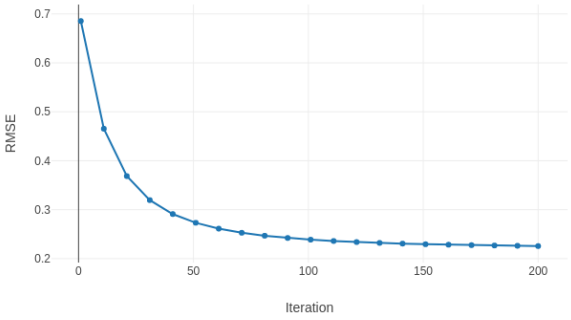

Abb. 7. Final model building withhelp of the total data set

1. (5.August 2018) https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf, MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS.