Advanced Recommender Systems – Home Assignment 1

Part 1 – SGD

1. Objective Function

$$\min_{x_u,\, y_i} \sum (r_{ui} - \hat{r}_{ui})^2 + \lambda \left[\sum_u \|x_u\|^2 + b_u^2 + \sum_i \|y_i\|^2 + b_i^2 \right]$$

2. Update Steps – Deriving the objective per parameter

$$\hat{r}_{ui} = x_u^T y_i + \mu + b_u + b_i$$

Xu

$$\frac{\partial}{\partial x_u} = -2 y_i \left[\overbrace{r_{ui} - \mu - b_u - b_i - x_u^T y_i}^{e_{ui}} \right] + 2\lambda_x x_u \qquad \backslash : 2$$

$$x_u^{New} = x_u^{old} + \eta \left(e_{ui} \cdot y_i - \lambda \cdot x_u \right)$$

Yi

$$\frac{\partial}{\partial y_i} = -2 x_u \left[\overbrace{r_{ui} - \mu - b_u - b_i - x_u^T y_i}^{e_{ui}} \right] + 2\lambda_y y_i \qquad \backslash : 2$$

$$y_i^{New} = y_i^{old} + \eta \left(e_{ui} \cdot x_u - \lambda \cdot y_i \right)$$

bi

$$\frac{\partial}{\partial b_i} = -2 \cdot \left(r_{ui} - \mu - b_u - b_i - x_u^T y_i \right) + 2 \cdot \lambda_i \cdot b_i$$

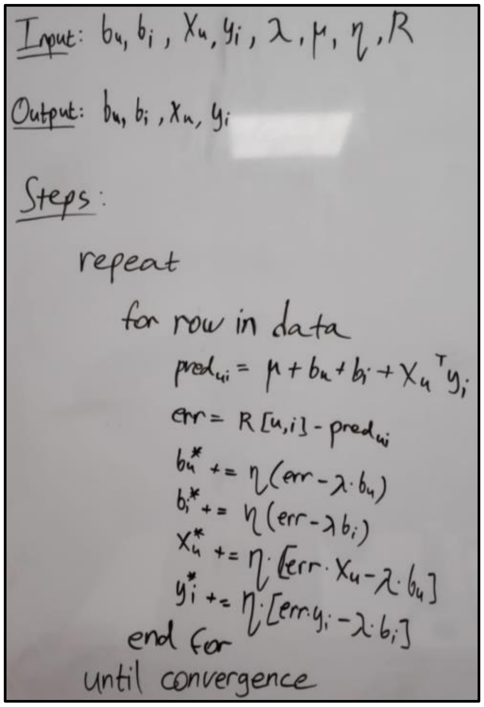
$$b_i^{New} = b_i^{old} + \eta \cdot \left(e_{ui} - \lambda \cdot b_i \right)$$

bu

$$\frac{\partial}{\partial b_u} = -2 \cdot \left(r_{ui} - \mu - b_u - b_i - x_u^T y_i \right) + 2 \cdot \lambda_u \cdot b_u$$

$$b_u^{New} = b_u^{old} + \eta \cdot \left(e_{ui} - \lambda \cdot b_u \right)$$

3. Pseudo Code



4. Hyper-parameters Tuning
   For SGD we will tune the parameters:
   a. K – Dimensions
   b. Eta – Learning rate
   c. Lambda – Regularization rate (for each individual parameter)
5. Working with the validation set and convergence tests
   After each iteration on the training set, we will check what is the RMSE on the validation set. Within the training set, we iterate over epochs. We defined convergence when the difference between epochs is smaller than 0.001 change <u>or</u> reaching the maximum number of epochs defined.
   We use the validation set to determine which hyper-parameter is the most optimal to use in our final model.
6. Training the best\last model
   By using the best model we found based on the training set and RMSE value on the validation set, we will then run it on both the training and validation sets combined. After convergence, we will use the model to run against the test set.
7. SGD Solution – main work items
   In order to achieve a better result we made a few steps in the code:
   a. Clipping – changing prediction results that are lower than one or higher than five.
   b. Hyperparameters Tuning – we used random search method which sample different parameters values in each iteration, in order to find the most suitable parameters for our training set.
8. Results (RMSE, MAE, R^2)
   RMSE is: 0.889745080993

MAE is: 0.6941523622081919
R^2 is: 0.363350125067

Part 2 – ALS

1. Objective Function

$$\min_{x_u,\, y_i} \sum (r_{ui} - \hat{r}_{ui})^2 + \lambda \left[\sum_u \|x_u\|^2 + b_u^2 + \sum_i \|y_i\|^2 + b_i^2 \right]$$

Exactly the same objective function as we used in the SGD model.

2. Update Steps – Deriving the objective per parameter

Preliminary step

$$\sum (r_{ui} - \hat{r}_{ui})^2 = \sum_D \Big(\overbrace{r_{ui} - \mu - b_u - b_i}^{a} - \overbrace{x_u^T y_i}^{b} \Big)^2 = \{a^2 - 2ab + b^2\}$$

$$\sum_D \left[(r_{ui} - \mu - b_u - b_i)^2 - 2 \cdot (r_{ui} - \mu - b_u - b_i) \cdot (x_u^T y_i) + x_u^T y_i \cdot y_i^T x_u \right] + \lambda \cdot \left(\sum_u \|x_u\|^2 + b_u^2 + \sum_i \|y_i\|^2 + b_i^2 \right)$$

Xu

$$\frac{\partial}{\partial x_u} = \sum_{u,i,r_{ui} \in D} \left[-2(r_{ui} - \mu - b_u - b_i) y_i + 2 \cdot y_i \cdot y_i^T x_u \right] + 2\lambda_x x_u = 0$$

$$\sum_D \left[y_i \cdot y_i^T x_u \right] + \overbrace{\lambda_x \cdot x_u}^{\lambda_x \cdot I \cdot x_u} = \sum_D \left[r_{ui} - \mu - b_u - b_i \right] \cdot y_i$$

$$\left[\sum_D \left[y_i y_i^T \right] + \lambda_x \cdot I \right] x_u = \sum_D \left[r_{ui} - \mu - b_u - b_i \right] y_i \quad \cdot \left(\sum_D [y_i y_i^T] + \lambda_x I \right)^{-1}$$

$$x_u^{New} = \left(\sum_D [y_i y_i^T] + \lambda \cdot I \right) \cdot \sum_D \left[r_{ui} - \mu - b_u - b_i \right] y_i$$

Yi

$$\frac{\partial}{\partial y_i} = \sum_{u,i,r_{ui} \in D} \left[-2(r_{ui} - \mu - b_u - b_i) x_u + 2 \cdot x_u \cdot x_u^T y_i \right] + 2\lambda_y y_i = 0$$

$$\sum_D \left[x_u \cdot x_u^T y_i \right] + \overbrace{\lambda_i \cdot y_i}^{\lambda_i \cdot I \cdot y_i} = \sum_D \left[r_{ui} - \mu - b_u - b_i \right] \cdot x_u$$

$$\left[\sum_D \left[x_u x_u^T \right] + \lambda_i \cdot I \right] y_i = \sum_D \left[r_{ui} - \mu - b_u - b_i \right] x_u \quad \cdot \left(\sum_D [x_u x_u^T] + \lambda_i \cdot I \right)^{-1}$$

$$y_i^{New} = \left(\sum_D [x_u x_u^T] + \lambda \cdot I \right)^{-1} \sum_D \left[r_{ui} - \mu - b_u - b_i \right] x_u$$

bu

$$\frac{\partial}{\partial b_u} = \sum_{D_u} 2\cdot\left(r_{ui} - \mu - b_i - x_u^T y_i + 2\cdot b_u\right) + 2\cdot\lambda_{b_u}\cdot b_u = 0$$

$$\sum_{D_u} b_u + \lambda_{b_u}\cdot b_u = \sum_{D_u}\left(r_{ui} - \mu - b_i - x_u^T y_i\right)$$

$$\left(|D_u| + \lambda_{b_u}\right) b_u = \sum_{D_u}\left(r_{ui} - \mu - b_i - x_u^T y_i\right) \quad \backslash \left(|D_u| + \lambda_{b_u}\right)^{-1}$$

$$b_u^{New} = \left(|D_u| + \lambda\right)^{-1} \cdot \sum_{D_u}\left(r_{ui} - \mu - b_i - x_u^T y_i\right)$$

bi

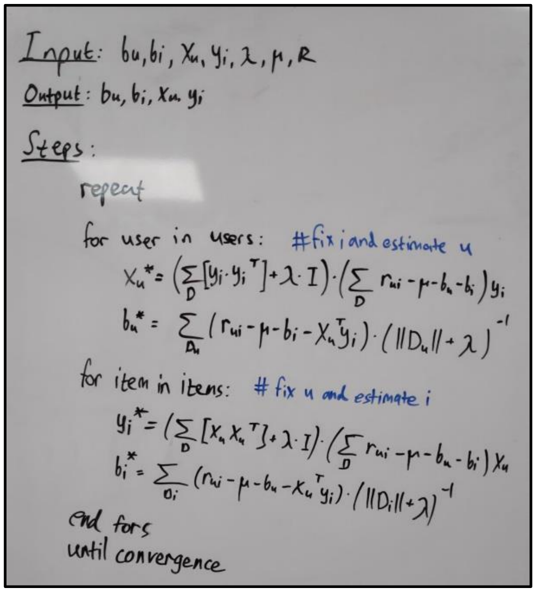
$$\frac{\partial}{\partial b_i} = \sum_{D_i} 2\cdot\left(r_{ui} - \mu - b_u - x_u^T y_i + 2\cdot b_i\right) + 2\cdot\lambda_{b_i}\cdot b_i = 0$$

$$\sum_{D_i} b_i + \lambda_{b_i}\cdot b_i = \sum_{D_i}\left(r_{ui} - \mu - b_u - x_u^T y_i\right)$$

$$\left(|D_i| + \lambda_{b_i}\right) b_i = \sum_{D_i}\left(r_{ui} - \mu - b_u - x_u^T y_i\right) \quad \backslash \left(|D_i| + \lambda_{b_i}\right)^{-1}$$

$$b_i^{New} = \left(|D_i| + \lambda\right)^{-1} \cdot \sum_{D_i}\left(r_{ui} - \mu - b_u - x_u^T y_i\right)$$

3. Pseudo Code



4. Hyperparameters Tuning:
    a. K – Dimensions
    b. Lambda – Regularization rate.
5. Working with the validation set and convergence tests
   Same as in SGD, after each iteration on the training set, we will check what is the RMSE on the validation set. We defined convergence when the difference between iteration is smaller than 0.001 change or reaching the maximum number of iterations defined.
6. Training the best\last model
   Same as SGD. By using the best model, we found based on the training set, we will then run it on both the training and validation sets combined. After convergence, we will use the model to run against the test set.
7. ALS Solution – main work items:
    a. Clipping – changing prediction results that are lower than one or higher than five.
    b. Hyperparameters Tuning – we used random search method which sample different parameters values in each iteration, in order to find the most suitable parameters for our training set
8. Results (RMSE, MAE, R^2)
   RMSE is: 0.9331830495897155
   MAE is: 0.7141202638038082
   R^2 is: 0.2996693236087469
9. Compare the ALS and SGD solutions in terms of implementation, training and quality.
   SGD
   Implementation – We first initialized the data into a pandas dataframe. Afterwards we created three main data structures to be used in the rest of the stages – two pandas series for the users and items, which was used to index all the available entities. In addition, we used pandas pivot table to recreate the original data into a matrix of users, items and their corresponding ratings.

Training – In each iteration we updated the latent matrices and the model parameters for all the users and items. We stopped iterating if the convergence condition has been satisfied, as described in the SGD section.
Quality – We will compare the models based on running time and efficiency. The SGD running times until the model converged were slightly longer, however the result received at the end were slightly better than the ones we received using the ALS model.
ALS
Implementation – We used the same structure as we did in the SGD model.
Training – On each iteration we first fixed the items latent matrix and updated the users parameters according to the fixed values, as a second step we did the exact same thing just changing the users to be fixed instead of the items.
Quality – As described in the SGD quality section above, the ALS model converged faster than the SGD one, but yielded a bit less results in terms of performance.