Name: Vigneshwaarar CR

CWID: A20392185

Bias Matrix Factorization

Matrix Factorization:

- In Matrix factorization, we learn much about the data, interaction between user and item. So, this is learning based method.
- Here, we create two new matrices named, User matrix and Item matrix. If suppose, we use K latent factors, M users and N items, then, User matrix will be MxK and Item matrix will be KxN
- So, multiplication of one row in user matrix with the one column in item matrix will give us the predicted rating for the user for the item. This can be represented as below,

$$X_{ui} = q^{T}{}_{i}\ p_{u}$$

- Initially, we assume very small value like 0.001 and form these two matrices. Now, we iterate through and update the values until the overall squared error is small.
- The formula for the updating the factors will be, here P is user matrix and Q is item matrix

$$P_i = P_i - \alpha \frac{1}{m} \sum_{k=1}^{m} \left((\mathrm{PiQj})^{(k)} - y^{(k)} \right)^{1} Q_i$$

$$Q_j = Q_j - \alpha \frac{1}{m} \sum_{k=1}^{m} \left((\mathrm{PiQj})^{(k)} - y^{(k)} \right)^{1} P_i$$

- In cases, there may be bais in user ratings for example a user might always give higher ratings i.e., his lowest rating might be 3 out of 5 for movies and few users might rate low always, i.e., his highest ratings might be low as 3. In those cases, we introduce user bias metric.
- Similarly, few items might always be rated high and few items might be always rated low, so to handle this item bias we introduce item bias metric.
- Our updated rating equation looks like,

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

Global average rating

Item bias

User bias

- We need to consider implicit feedbacks which improves our prediction accuracy, i.e., if a user has rated an item it means that he is interested in the item and if he didn't rate it might mean that he is not that interested the item. In these cases, we introduce a metrics for implicit feedbacks.

- User attributes are also considered, while building our model, because, attributes of the users might help us in predicting the values. Thus, we update the loss function with the metrics as shown below.

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T [p_u + |N(u)|^{-0.5} \sum_{i \in N(u)} x_i + \sum_{a \in A(u)} y_a]$$

We extend the p_u vector in this way

Implicit feedbacks

User attributes

- We also introduce regularization terms like Lazzo(L1 term) and Ridge terms to our cost function

$$L1 = \lambda|P_i| + \lambda|Q_j|$$

$$\text{L2 regularization term} = \frac{\lambda}{2}||P_i||^2 \ \frac{\lambda}{2}||Q_j||^2$$

- Here, we can evaluate using metrics like Mean Absolute Error

Execution steps:

NOTE: PLACE THE INPUT FILES NAMED “train.txt” and “test.txt” IN THE SAME FOLDER AS THE .jar.

- Run the .jar file as following command format
  java -jar mf.jar -e 0 -k 100 -lr 0.01 -rr 0.001
- Enter the biased MF as 0 or biased MF with extension as based 1
- Enter k value next, which can be any integer
- Next inter lr value, which means learning rate
- Regularization rate rr is the next value to be entered.

MAE for different values: