DESIGN DOCUMENTATION

This assignment evaluates the speed and accuracy of different recommendation systems given below

1. Collaborative filtering
2. Collaborative along with Baseline approach
3. SVD
4. SVD with 90% retained energy
5. CUR
6. CUR with 90% retained energy

This assignment is divided into 7 subcodes.

- svd.py
- collaborative_filtering.py
- cur.py
- data_handling.py
- error_funcs.py
- similarity_funcs.py
- final_run.py

svd.py

In this method a data matrix(A) is divided into three sub matrices, using singular value decomposition.

A = U * Sig * V'

A = Original Data Matrix (Users * Items)

U = Users to Concept matrix

V = Items to Concept matrix

Sig = Concept Strength matrix containing Eigen values in decreasing order

Here this part of the code takes the A matrix with few values removed and it tries predicting it. Now the error between the predicted values and the original values will be compared.

collaborative_filtering.py

This method implements an approach where we take ratings of users similar to the one we need to predict, from A matrix and estimate a value based on weighted similarity of their ratings. Here we take two similar users but it may vary as per the requirement. In baseline approach we add a new term b, while estimating the weighted similarity, where b is the sum of average of all the ratings, deviation of the user and deviation of the item.

cur.py

Here A matrix is split into C, U, R matrices where C contains of few columns randomly picked from matrix A. U is a matrix which is generated through a particular algorithm and R contains few rows which are randomly picked from A.

A = C*U*R

Where, C =column matrix,

U = pseudo inverse of intersection of C and R,

R = row matrix.

data_handling.py

It will retrieve the data from the dataset which is of the form user-id, movie-id, rating, time-stamp as four columns in it. From this four columns user item matrix is created which is used in the recommendation system

error_funcs.py

This part of the code contains three different measures to find the accuracy of recommendation systems:

1. RMSE - Root Mean Square Error
2. Precision in Top K
3. Spearman Correlation

RMSE - Root Mean Square Error:

Formula: (sum((predicted - actual) ** 2) / n) ^ 0.5

Precision in Top K:

Gives an estimate of how many of the predicted ratings are present in the top K ratings of the user since only the good one's count in the error measure.

Spearman Correlation:

Formula: 1 - [sum(diff(predicted - actual)^2) / n((n^2)-1)]

similarity_funcs.py

In this part of the code pearson similarity for given two matrices will be calculated

$$\boldsymbol{sim(x,y)} = \frac{\sum_{s\in S_{xy}}(\boldsymbol{r_{xs}} - \overline{\boldsymbol{r_x}})(\boldsymbol{r_{ys}} - \overline{\boldsymbol{r_y}})}{\sqrt{\sum_{s\in S_{xy}}(\boldsymbol{r_{xs}} - \overline{\boldsymbol{r_x}})^2}\sqrt{\sum_{s\in S_{xy}}(\boldsymbol{r_{ys}} - \overline{\boldsymbol{r_y}})^2}}$$

final_run.py

This is a control unit for the whole code, which combines all the functions written above in an appropriate way and gives output.

Data_set

All the data here is stored in the folder dataset which contains a file namely u.data. In this file, data of 10^5 ratings are stored which is collected from internet. In this file data is stored in four columns, where 1st column contains user id's, 2nd column contains movie id, 3rd column contains rating given by the specific user and 4th column contains the time stamp.