Assignment 2 - Design Document

Recommender Systems

Vijitha Gunta (2015B3A70491H)

Shivani Reddy (2015B3A70531H)

Shubham Bharadwaj (2016A7PS0120H)

INTRODUCTION

To implement and compare various techniques for building a Recommender System. The techniques implemented are SVD, SVD with 90% retained energy, CUR, CUR with 90% retained energy, collaborative filtering and collaborative along with baseline approach.

Problem Desc.

To apply several techniques of recommender system on a user-movie rating matrix to predict ratings and then compare their results.

Technologies Used

1. Python 3 + dependencies (numpy, timeit, math, scipy, sklearn, random)
2. Anaconda
3. Jupyter notebook
4. Movie - User rating test data set

Dataset Used

MovieLens 100K dataset is used which consists of 100,000 ratings from 1000 users on 1700 movies. The dataset is available online on https://grouplens.org/datasets/movielens/100k/ and can be downloaded from there. 'Process' function found in svd and cur files does the preprocessing of the .txt file (pre-processed from the csv file, keeping only relevant data which is users, movies and ratings and excluding time-stamp) and constructs the user-movie utility matrix. This matrix is then modified by subtracting mean of every row from every entry in row to handle bias of harsh and lenient users as explained in detail later. It returns this normalized matrix (numpy array) which is then worked upon by all other functions.

COMPARITIVE STUDY

Recommender System	RMSE	TOP K PRECISION	SPEARMAN CORRELATION	TIME TAKEN
Collaborative	1.748	75.62	0.9999	2532.67
Collab + baseline	1.296	79.65	0.999985	0.999985
SVD	1.116	66.881	0.9999978	91.528
SVD 90% energy	1.076	76.295	0.999999	87.697
CUR	3.3069	87.697	0.99999	200.996

CUR 90% energy	1.088	58.2447	0.99999	209.699

Collaborative Filtering

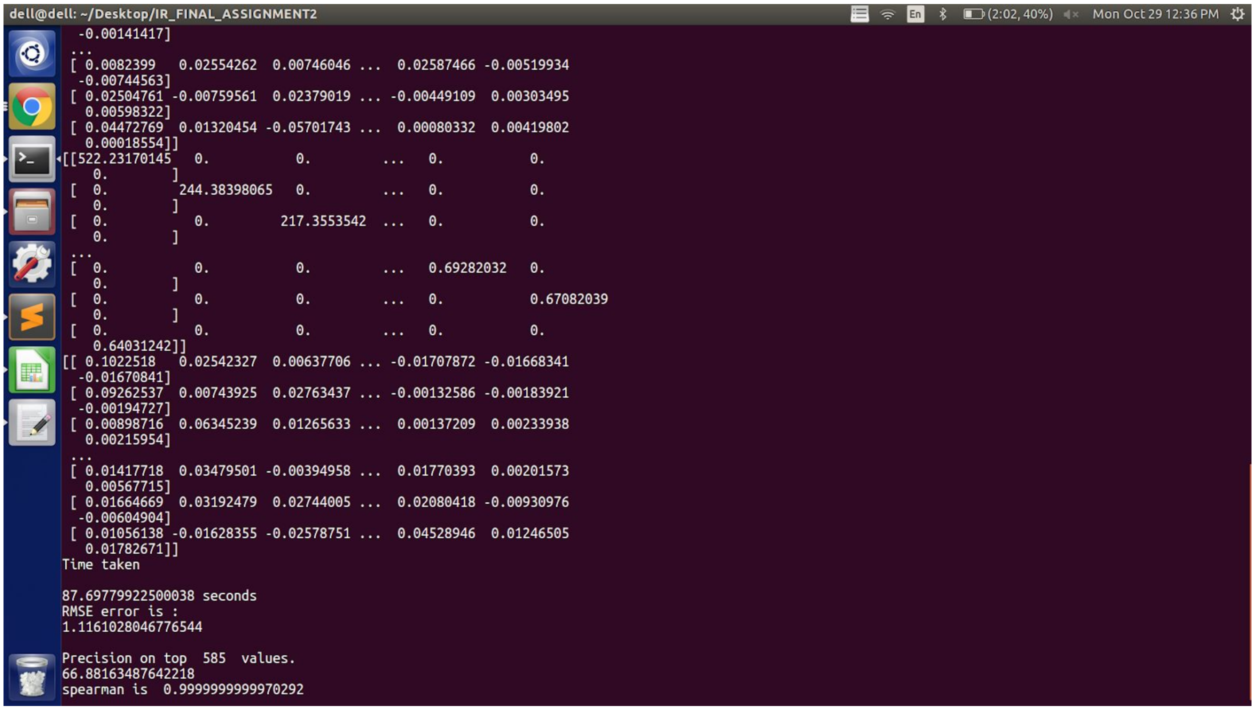
Collaborative filtering filters information by using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. A person who wants to see a movie for example, might ask for recommendations from friends. The recommendations of some friends who have similar interests are trusted more than recommendations from others. In the neighbourhood-based approach, a number of users are selected based on their similarity to the active user. A prediction for the active user is made by calculating a weighted average of the ratings of the selected users.

Collaborative Filtering with baseline

This technique is the same as collaborative filtering except that the ratings are modified according to a baseline approach. The ratings are normalized by adjusting with global average of all ratings and then adjusting with the local average rating of the user. After this is done, the general steps of collaborative filtering are followed for prediction.

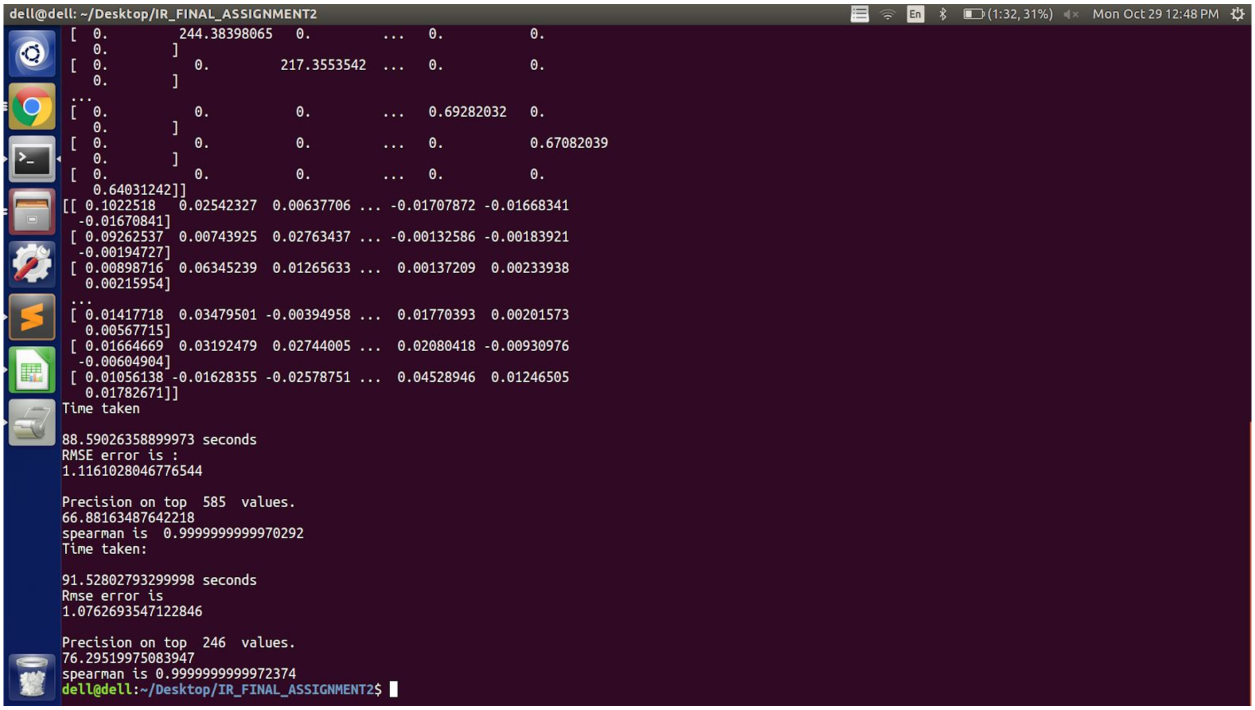
SVD

SVD decomposes the initial user-movie rating matrix into three matrices for dimensionality reduction. These three matrices (U, sigma and V) are calculated on the basis of eigenvalues and eigenvectors of the original matrix. The eigenvectors should give out a positive eigenvalue, otherwise it is multiplies with minus one. These matrices are then multiplied by dot product to give the predicted ratings.



SVD with 90% retention

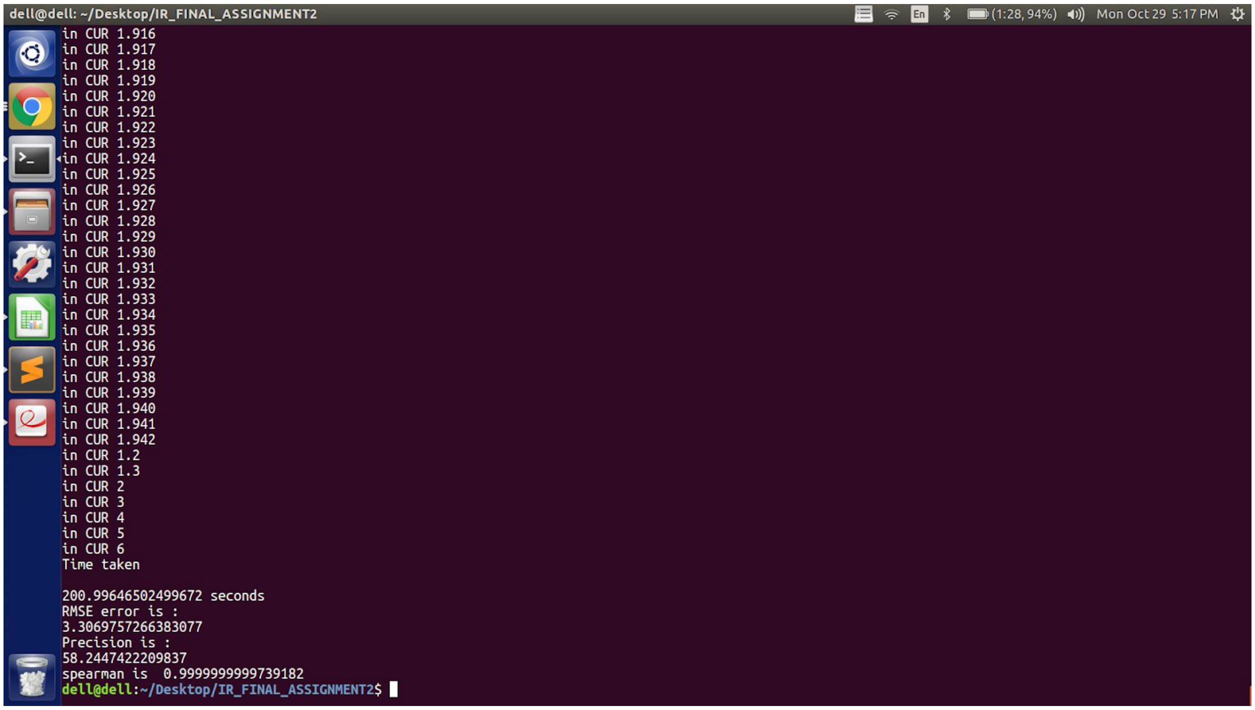
It follows the same steps as SVD but takes only some eigenvalues into account in the sigma matrix. Sigma matrix has descending order of eigenvalues in its diagonal. The smallest of these are removed and corresponding changes are made to U and V as well so that dot product is not a problem. Their removal is based upon the fact that at least 90% energy should be retained, that is, the sum of squares of remaining eigenvalues should at least be 90% of the original sum of squares of all eigenvalues. After that, the general procedure of SVD prediction is followed for prediction.



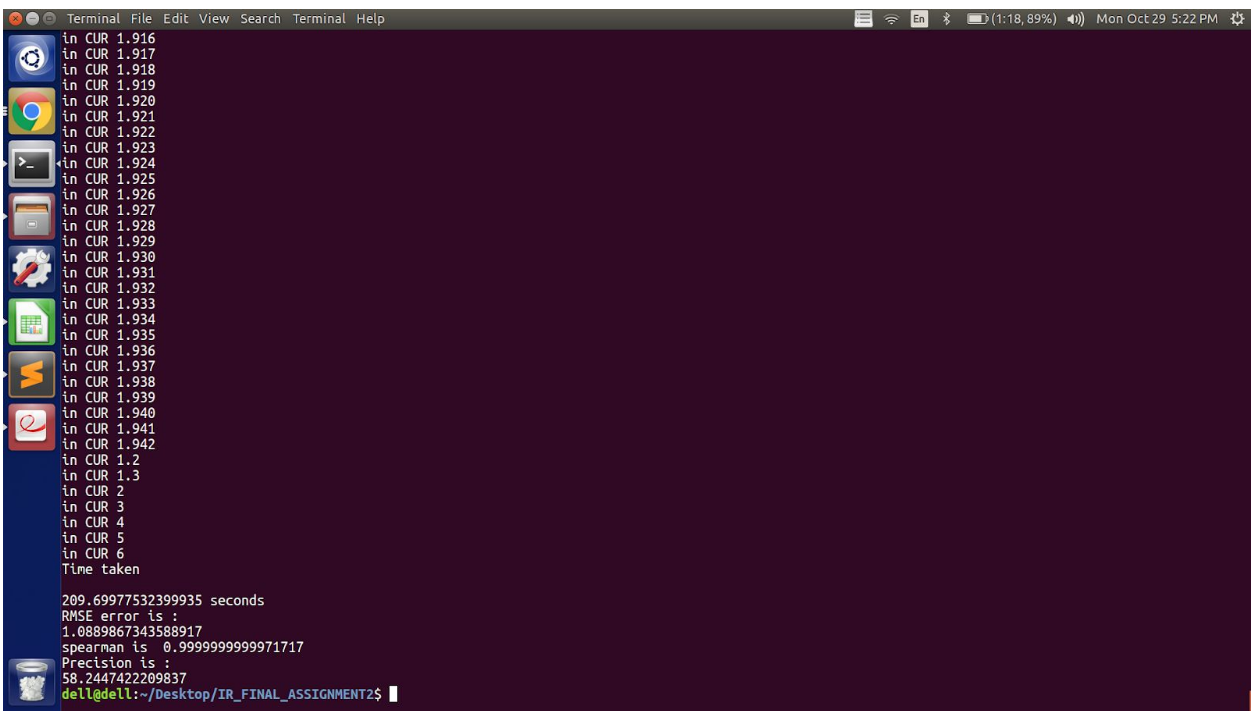
CUR

This is also a decomposition procedure for matrices. A CUR matrix approximation of the original matrix is three matrices C, U and R such that C is made from columns of original matrix, R is made from rows and that the product CUR is a close approximation of ratings matrix.

In this implementation of CUR we took 'r' the approximation rank to be the rank of original matrix. Rows and columns are selected in a random fashion using numpy function and we avoid dealing with duplicates by specifying to function as parameter.



CUR with 90% retention



REFERENCES

1. Corpus: https://grouplens.org/datasets/movielens/100k/
2. pydoc: https://www.youtube.com/watch?v=Y6TgbyfKCNM
3. https://medium.com/@m_n_malaeb/recall-and-precision-at-k-for-recommender-systems-618483226c54