BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

HYDERABAD CAMPUS

Information Retrieval (CS F 469)

Assignment 2

First Semester 2016-2017

Members	ID No.
PRASANTH YADLA	2013B5A7561H
AMAN NIDHI	2013A3PS400H

Prepared under:

Dr. Aruna Malapati

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

HYDERABAD CAMPUS

(November 2016)

SVD vs CUR DECOMPOSITION

Language Used: Python3.5

API/Tools Used: NumPy, SciPy Scientific Computing Packages

This assignment consists of two parts.

Part1 : Calculating Singular Valued Decomposition (SVD)

1. Selection of dataset : Movielens from grouplens.com containing data from 672 users on 9066 movies.
   The dataset we used consists of user-movie ratings. Each user would give ratings for few movies they watched. So the corresponding matrix can be sparse. To test the program add the csv file to the path.
2. SVD Algorithm: The algorithm consists of decomposing a given matrix into 3 smaller size matrices

$$A = U \in V^T$$

   Where,
   A is input matrix,
   U is user-to-concept matrix
   ∈ is strength of corresponding concept
   V is movie-to-concept matrix
3. Implementation: U is calculated using the normalized eigenvectors for the matrix ($A*A^T$) .This is done in base.py file as a function eighk(M,k). Similarly V is calculated using the eigenvectors of (A^T*A).The concept strength matrix is just the square root of eigenvalues.

We also classify as to whether matrix is sparse or not and build functions accordingly.
We are also reading the data with pandas and porting the panda's dataframe to NumPy array for fast matrix calculations.

4. Data structure used: list of lists (matrix).The input data matrix has rows as users and columns as movies and their corresponding matrix entry as rating provided by the user for that particular movie.

Part2: Calculating CUR decomposition

1. Dataset: Same dataset as SVD
2. CUR Algorithm: This method picks randomly a set of columns C and rows R. U is constructed using the pseudo-inverse of the intersection of the chosen rows and columns.

$$A = CUR$$

Where,
A is input matrix
C is sample columns
R is sample rows
Unlike SVD, CUR has C and R which are Big but Sparse, whereas U is dense and small.

3.Implementation:

The function CURCalculate(A,samplerow,samplecol) is called first

specified with actual parameters by the programmer. It caluculates the probability distributions for each row, column and is sampled by the function choose_rand(probarray).Intersection is calculated and U matrix is calculated according to the algorithm.

Finally after a given number of runs , the average frobenius error is calculated and printed for the matrix

4.Data structure used: list of lists (matrix).The input data matrix has rows as users and columns as movies and their corresponding matrix entry as rating provided by the user for that particular movie.

Analysis:

SVD working on the data took ~2.5 seconds whereas CUR algorithm took ~ 0.3 seconds on the same dataset, i5 processor , 4GB RAM.

The Frobenius Norm of SVD was approaching 0 compared to CUR which was very high maybe because of unlucky probability sampling.