INFORMATION RETRIEVAL

CS F469

RECOMMENDER SYSTEMS

Pranjal Gupta	2017A7PS0124H
Daksh Yashlaha	2017A7PS0218H
Satyam Mani	2017A7PS0277H
Ujjwal Raizada	2017A7PS1398H

Few important details

- Dataset used: MovieLens dataset
- Strict and easy raters are handled properly using pearson correlation
- CUR details
    - Rows -
    - Columns -

Recommender System Technique	**Root Mean Squared Error (RMSE)**	**Mean Average Error (MAE)**	**Time taken for prediction**
Collaborative	1.15810598485930 6	0.59734323668047	12.69501471519s
Collaborative along with Baseline approach	0.59405799171546	0.511784030852735	8.78501643517s
SVD	0.89123464828352	0.854693215875624	12.8539641275s
SVD with 90 per cent energy	0.93576824631325	0.902395864782184	10.6939641351s
CUR	0.98564682842824	0.945845754924922	12.641667522s
CUR with 90 per cent energy	1.04542854655445	0.965846427454562	12.2184527496s
Latent factor model	0.89146426426542	0.862584645656546	12.8454651465s

- Language Used : python3
- Libraries used
    - Numpy
    - Pandas
    - Scipy
    - Sklearn

Collaborative Filtering and filtering with baseline approach

The process of identifying similar users and recommending what similar users like is called collaborative filtering.

Similarity Measure used: Pearson correlation

Major Data-Structures used:

- df : pandas dataFrame
    - index (along the row): user
    - label (columns) : movies
- utility_matrix : 2d numpy array
    - row - user_id
    - Column - movie_id
- Users_map: { key, value }
    - key : user_no
    - value : iterator in the range(len(user_map))
- Movie_map: { key, value }
    - key : movie_no
    - value : iterator in the range(len(movie_map))

*** Same data-structures were followed for the baseline approach ***

Singular Value Decomposition

SVD: SVD is calculated for any given MxN matrix by decomposing that matrix into three component matrices defined as follows:

- Matrix U: An Mxr column orthonormal matrix where r is the rank of the original matrix.
- Matrix $\sum$: An rxr diagonal matrix containing the singular values of the original matrix.
- Matrix V: An rxN column orthonormal matrix where r is the rank of the original matrix.

The original matrix can then be approximated by calculating the product of these matrices as follows:
Approximation of original matrix = $U*\sum*V^T$

Dimensionality Reduction:
If the rank r of the matrix is substantially smaller than N, we can intuitively see that the combined sizes of the component matrices will be smaller than that of the original matrix, thereby providing a valuable reduction in dimensionality for computational purposes.

90% energy rule: Retain enough singular values to make up 90% of the energy. That is, the sum of squares of the retained singular values should be at least 90% of the sum of squares of all the singular values.

CUR Method and CUR with 80% - 90% energy

Utility matrix is divided into three matrices C, U, R

Major Data Structure:

- Matrix C: An Mxr column orthonormal matrix where r is the rank of the original matrix.
- Matrix U: An rxr diagonal matrix containing the singular values of the original matrix.
- Matrix R: An rxN column orthonormal matrix where r is the rank of the original matrix.

The original matrix can then be approximated by calculating the product of these matrices as follows:
Approximation of original matrix = $C * U * R^T$

LATENT FACTOR MODEL

In latent factor based method, we only feed user's history and we need not to define descriptors or factors. The algorithm will find the hidden factors that influence the user's preference like brand of product, price of product and so on for us. Here, no product description is required, only user's history is good.

Major Data Structures Used:

- utility_matrix: 2d array
    - row - user_id
    - columns - movie_id
- user_factor_matrix : 2d array
    - Computed after decomposition of utility matrix into latent factors
- product_factor_matrix : 2d array
    - Second result due to decomposition of utility matrix into latent factors