Information Retrieval

Assignment 3

Recommender System

Developers:

Rishab Nahar-2018A7PS0173H
Nipun Wahi-2018A7PSH
Samkit Jain-2017BA7H
Aditya Pandey-2018A3PS0517H

Objective:

This assignment involved the development of a mini recommender system like the one which Netflix uses, but on a smaller dataset. The aim of this assignment was to predict movie ratings of an unknown user given his history and ratings of other users and test the model by Root Mean Square Error, Precision on top K and Spearman Rank Correlation.

We are working on a dataset which contains contain 1,000,000 anonymous ratings of approximately 2400 movies made by 1600 users.The objective is to predict the rating of a user by various models like Collaborative Filtering, Collaborative filtering with baseline approach, SVD, CUR.

Packages/Libraries Used

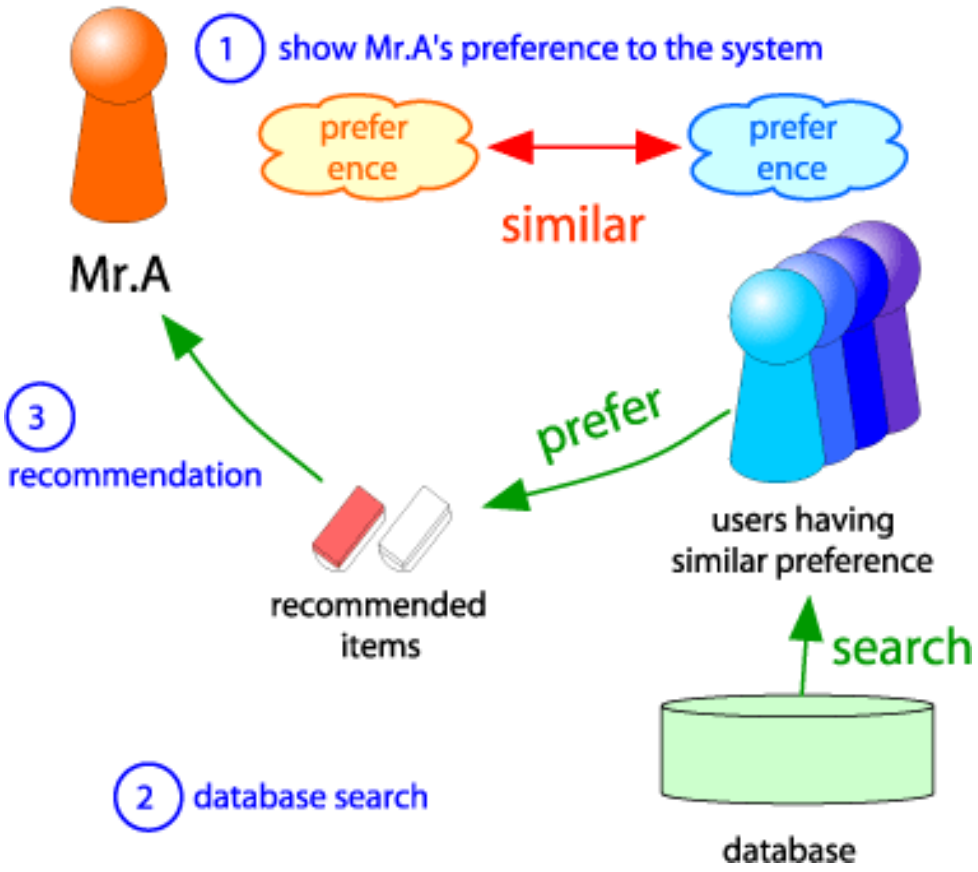
1) Pandas for data distribution
2) Numpy for all the matrix airthmetics
3) Math
4) Sk Learn for splitting data into 2 parts

Data Structures Used

1) Numpy Array
2) Lists

Collaborative Filtering

Design Architecture for Collaborative filtering



An illustration showing the major steps in Collaborative Filtering

The main steps involved in user-user collaborative filtering involve finding similar users like target user X and using

cosine similarity between them to predict the rating for target user X based on the rating of similar users.

Item-item collaborative filtering uses a similar approach but finds similarity between movies instead of users. By training the model on various datasets it has been found that item-item collaborative filtering gives better results as compared to user-user collaborative filtering.

We have considered cosine distance as the distance measure to find similarity.

Data pre-processing:

The first step before creating a dataframe involved normalising the values given in ‘u.data’.

We normalised the values to ensure that strict and generous raters are treated fairly. We subtracted the mean rating from every user rating to ensure this fairness.Those users who did not rate a movie X , we avoided subtracting the mean.

We use a NUMPY array for storing the whole data. We divided the whole dataset into two parts one for testing and the other for training. The train matrix can be termed as utility matrix.

Prediction of Ratings:

Once we have normalized utility matrix, we can use it for predicting ratings based on user-user and item-item collaborative filtering methods.

For every unknown rating in our testing data, we compute first the similarity of all movies in item-item collab filtering, take out the ones having positive similarity and compute the predicted rating using weighted average.

$$r_{xi} = \frac{\sum_{j \in N(i;x)} s_{ij} \cdot r_{xj}}{\sum_{j \in N(i;x)} s_{ij}}$$

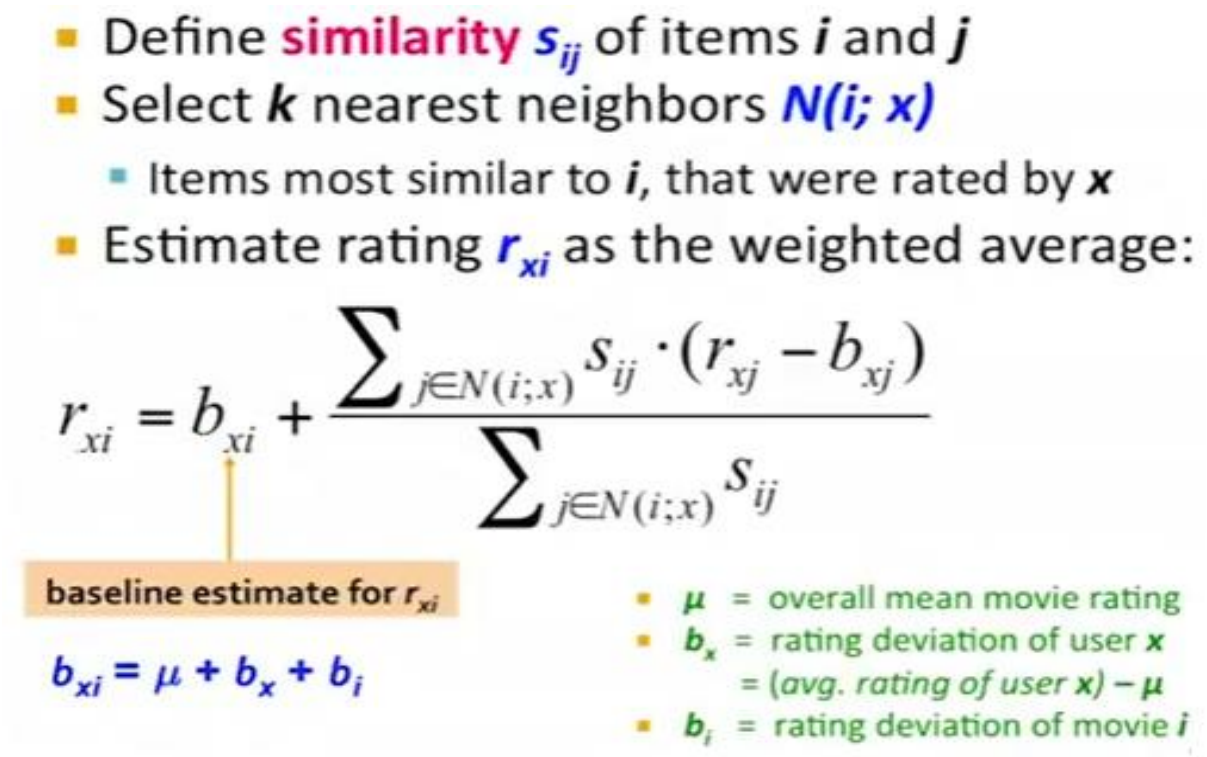
Where

s_{ij}… similarity of items ***i*** and ***j***
r_{xj}…rating of user ***u*** on item ***j***
N(i;x)… set items rated by ***x*** similar to ***i***

and

$$sim(x, y) = \frac{\sum_i r_{xi} \cdot r_{yi}}{\sqrt{\sum_i r_{xi}^2} \cdot \sqrt{\sum_i r_{yi}^2}}$$

Baseline approach:



Baseline approach tries to incorporate the strict and generous raters scenario, by adding the bias of each user x and movie. This improves our model to some extent because strict and generous raters are handled precisely.

Results:

Collaborative Filtering

The Top 35 neighbours were considered for finding the similarity between the unknown user and known user

The Root mean Square error in collaborative filtering model without baseline approach is 1.024662168682888

The Prescison at top K in collaborative filtering model without baseline approach is 0.6648250460405156

The Spearman Rank Correlation in collaborative filtering model without baseline approach is 0.9694352854352855

Time taken by the above model is 11.138118267059326

Collaborative Filtering with Baseline approach

The Top 35 neighbours were considered for finding the similarity between the unknown user and known user

The Root mean Square error in collaborative filtering model with baseline approach is 0.953464891490556

The Prescison at top K in collaborative filtering model with baseline approach is 0.6777163904235728

The Spearman Rank Correlation at top K in collaborative filtering model with baseline approach is 0.9773792633792634

Time taken by the above model is 12.476200103759766

Pros and Cons of Collaborative Filtering

Pros: -

1) No domain knowledge necessary
2) The model can help users discover new interests.
3) To some extent, the system needs only the feedback matrix to train a matrix factorization model.

Cons: -

1) Cannot handle fresh items
2) Hard to include side features for query/item
3) Latent factors are not included in collaborative filtering approach

Single Value Decomposition

Singular value decomposition takes a rectangular matrix of gene expression data (defined as A, where A is a *n* x *p* matrix) in which the *n* rows represents the genes,and the *p* columns represents the experimental conditions.

The SVD theorem states:

$$\mathbf{A}_{nxp} = \mathbf{U}_{nxn}\,\mathbf{S}_{nxp}\,\mathbf{V}^{T}_{pxp}$$

Where
$\mathbf{U}^{T}\mathbf{U} = \mathbf{I}_{nxn}$
$\mathbf{V}^{T}\mathbf{V} = \mathbf{I}_{pxp}$ (i.e. U and V are orthogonal)

Where the columns of U are the left singular vectors (*gene coefficient vectors*); S (the same dimensions as *A*) has singular values and is diagonal (*mode amplitudes*); and V^{T} has rows that are the right singular vectors (*expression level vectors*). The SVD represents an expansion of the original data in a coordinate system where the covariance matrix is diagonal.

Results for SVD:

The results for 100% retained energy

The Root mean Square error in SVD model with 100% retained Energy is 0.6620215808837481

The Prescison at top K in SVD model with 100% retained Energy is 0.5292242295430393

The Spearman Rank Correlation at top K in SVD model with 100% retained Energy is 0.8973331696052059

Time taken by the above model is 8.574554920196533

The results for 90% retained energy

The Root mean Square error in SVD model with 90* retained Energy is 0.6471290530964965

The Prescison at top K in SVD model with 90* retained Energy is 0.5302226935312832

The Spearman Rank Correlation at top K in SVD model with 90* retained Energy is 0.9093341732675948

Time taken by the above model is 8.537238121032715

Pros and Cons of SVD

Pros: -

1) Simplifies data and in some ways shows the latent factors
2) Removes Noise
3) Best algorithm for finding recommendations

Cons: -

1) Transformed data may be difficult to understand
2) Highly expensive in terms of space and time complexity

CUR

A **CUR matrix approximation** is a set of three matrices that, when multiplied together, closely approximate a given matrix. A CUR approximation can be used in the same way as the low-rank-approximations of the Singular Value Decompositions (SVD). CUR approximations are less accurate than the SVD, but they

offer two key advantages, both stemming from the fact that the rows and columns come from the original matrix (rather than left and right singular vectors):

1)There are methods to calculate it with lower asymptotic time complexity versus the SVD.

2)The matrices are more interpretable; The meanings of rows and columns in the decomposed matrix are essentially the same as their meanings in the original matrix.

Formally, a CUR matrix approximation of a matrix *A* is three matrices *C*, *U*, and *R* such that *C* is made from columns of *A*, *R* is made from rows of *A*, and that the product *CUR* closely approximates *A*. Usually the CUR is selected to be a rank-*k* approximation, which means that *C* contains *k* columns of *A*, *R* contains *k* rows of *A*, and *U* is a *k*-by-*k* matrix. There are many possible CUR matrix approximations, and many CUR matrix approximations for a given rank.

Algorithm for CUR

Let M be a matrix of m rows and n columns. Pick a target number of "concepts" r to be used in the decomposition. A CUR-decomposition of M is a randomly chosen set of r columns of M, which form the m × r matrix C, and a randomly chosen set of r rows of M, which form the r × n matrix R. There is also an r × r matrix U that is constructed from C and R as follows:

1. Let W be the r × r matrix that is the intersection of the chosen columns of C and the chosen rows of R. That is, the element in row i and column j of W is the element of M whose column is the jth column of C and whose row is the ith row of R.

2. Compute the SVD of W; say W = XΣY T.

3. Compute Σ+, the Moore-Penrose pseudoinverse of the diagonal matrix Σ. That is, if the ith diagonal element of Σ is σ

6= 0, then replace it by 1/σ. But if the ith element is 0, leave it as 0.

4. Let U = Y (Σ+) 2XT.

Results for CUR:

The results for 100% retained energy

300 Rows and Column were taken in CUR for calculation.

The Root mean Square error in CUR model with 100% retained Energy is 0.720620129684795

The Prescison at top K in CUR model with 100% retained Energy is 0.3273092369477912

The Spearman Rank Correlation at top K in CUR model with 100% retained Energy is 0.918654248001577

Time taken by the above model is 1.5806362628936768

The results for 90% retained energy

300 Rows and Column were taken in CUR for calculation.

The Root mean Square error in CUR model with 90% retained Energy is 0.8088241441403514

The Prescison at top K in CUR model with 90% retained Energy is 0.33178500331785005

The Spearman Rank Correlation at top K in CUR model with 90% retained Energy is 0.9734934071742437

Time taken by the above model is 1.5179753303527832

Pros and Cons of CUR

Pros: -

1) This is much more similar to SVD with approximately same results and accuracy.
2) This is less expensive in terms of memory and time complexities.

Cons: -

1) While choosing random rows and columns reptations can come which will result in extra weight of a particular movie

Finally we can sum up our results in a table as follows:-

Recommender System Technique	**Root Mean Square Error (RMSE)**	**Precision on top K**	**Spearman Rank Correlation**	**Time taken for prediction (in sec)**
Collaborative	**1.024**	**0.665**	**0.970**	**11.138**
Collaborative along with Baseline approach*	**0.953**	**0.677**	**0.977**	**12.476**
SVD	**0.662**	**0.529**	**0.897**	**8.574**
SVD with 90% retained energy	**0.647**	**0.530**	**0.909**	**8.537**
CUR	**0.721**	**0.327**	**0.918**	**1.580**
CUR with 90% retained energy**	**0.808**	**0.331**	**0.973**	**1.517**