Design Document

Introduction

THE ASSIGNMENT:

This Assignment has been worked on by members of the team, on **Dimensional Reduction methods for Recommender Systems**, using two different dimensional reduction methods- **SVD** and **CUR**.

The purpose of this design document is to explore the details of the ideas behind the code-about SVD and CUR
and how the software works as an implementation of Recommender System. We have chosen to code the dimensional methods of SVD and CUR for the data set pertaining to the data set "MovieLens", provided by the GroupLens Research association from the MovieLens website. This data was collected over a period of time, with about 100000 ratings provided for 9000 movies by 700 users with about 1300 tag applications.

RECOMMENDER SYSTEMS- A BRIEF INTRO

There are various web-based applications that enable one to choose from a variety of products or items based on the predictions made using the user's previously chosen options and options of other users. These are known as Recommender Systems.
There are different kinds of Recommender System, which can be broadly divided into two different categories:

- **Content-Based Systems**
- **Collaborative Filtering Systems**

Here we will look into the Collaborative Filtering System- and the decomposition methods in particular.

DIMENSIONAL REDUCTION:

Manifold learning is a significant problem across a wide variety of information processing fields including pattern recognition, data compression, machine learning, and database navigation. In many problems, the measured data vectors are high-dimensional.

may have reason to believe that the data lie near a lower-dimensional manifold Learning a suitable low-dimensional manifold from high-dimensional data is essentially the same as learning
Dimensionality reduction can also be seen as the process of deriving a set of degrees of freedom which can be used to reproduce most of the variability of a data set.
Consider a set of images produced by the rotation of a face through different angles. Clearly only one degree of freedom is being altered, and thus the images lie along a continuous one-dimensional curve through image space.

SVD - Singular Value Decomposition

The singular value decomposition (SVD) is a generalization of the eigen decomposition which can be used to analyze rectangular matrices (the eigen decomposition is defined only for squared matrices). By analogy with the eigen decomposition, which decomposes a matrix into two simple matrices.

The main idea of SVD is to decompose the given matrix into three different matrices. These three matrices are:
2 Orthogonal Matrices - U and V^T (transpose of V)
1 Diagonal Matrix - Sigma / K

This matrix is obviously square and symmetric, but also (and this is less obvious) its eigenvalues are all positive or null, and the eigenvectors corresponding to different eigenvalues are pairwise orthogonal.

Formally, if A is a rectangular matrix, SVD decomposition occurs as

```
A = UKV^T
```

with:
A →(**mxn** matrix)
U → **(mxr** matrix)the normalized eigen vectors of the matrix AA^T, the columns of U are called left-singular vectors of A.
V →(**nxr** matrix) the normalized eigen vactors of the matrix A^T A, the columns of V are called right-singular vectors of A.
K →(**rxr** matrix) the diagonal matrix of the singular values, K = L^1/2 where L being the diagonal matrix of the eigen values of the matrix AA^T and A^TA

SVD - PROPERTIES:

It is possible to decompose the matrix into the above mentioned three matrices, such that

1. All three are unique
2. U and V are column orthogonal and dense
3. K is diagonal and sparse, such that the entries are positive and are sorted in a decreasing order.

CUR Decomposition:

It is a low-rank approximation explicitly expressed in terms of a small number of columns and rows of the original matrix A:

```
A = C U R
```

Here,
A →the matrix (mxn)
C →is made from the columns of A (mxr)
R →matrix taken from the rows of A (rxn)
U →An rxr matrix, that is calculated using the SVD technique.

There are certain drawbacks with using SVD as a dimensional reduction technique, which can be improved using CUR:

- **More Time:** SVD takes more time to process, and while SVD guarantees optimality, time is of the essence
- **More Computational Cost:** SVD while optimal, here we can decrease the computational cost, while allowing for an increase in error.

CUR - PROPERTIES:

1. **Easy to Interpret**
2. **Sparse Basis Vectors of Matrix**

3. **Duplicate columns and rows**

Here U is a matrix that is constructed by a pseudo-inverse of the intersection of the chosen rows and columns from the matrix A.

LATENT FACTOR MODEL:

Latent factor models are an alternative approach that tries to explain the ratings by characterizing both items and users. It is based on matrix factorization.

The model uses a stochastic gradient method for optimization.

- $q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$
- $p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u)$

The system minimizes the regularized squared error on the set of known ratings:

$$\min_{p\cdot,q\cdot,b\cdot} \sum_{(u,i)\in\kappa} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda$$
$$(\| p_u \|^2 + \| q_i \|^2 + b_u^{\,2} + b_i^2)$$

The system learns the model by fitting the previously observed ratings. Thus, the system should avoid overfitting the observed data by regularizing the learned parameters, whose magnitudes are penalized.

OBSERVATIONS:

Recommender System Technique	**Root Mean Squared Error (RMSE)**	**Mean Average Error (MAE)**	**Time taken for prediction**
Collaborative	0.8415866539574305	0.656756870276191	9.534794092178345
Collaborative along with Baseline approach	0.8792696706813906	0.6832819991715989	20.99986624717712
SVD	0.011385705026808059	3.5328357027018216	3.510614156723022

SVD with 90 per cent energy	0.011399370975422835	3.533864176046703	2.373655319213867
CUR	0.20654564810588275	9.436670770200255	5.768579244613647
CUR with 90 per cent energy	0.01171624513229387	3.52986	5.761593341827393
Latent factor model	1.1228	0.9404	3.234384298324585

Conclusion:

In our understanding of CUR and SVD, we come across the fact that the reconstruction error of CUR is less than or equal to that of SVD plus some (epsilon) times of the Frobenius norm of A. Also, it can be noted that it takes less time to process the code for CUR in comparison for SVD.

While it can be argued that either of SVD or CUR are better or not, we have chosen to provide implementations of both in our software.