Design Document

Team Members

1) Ansuman Dibyayoti Mohanty (2016A7PS0043H)
2) Syed Abid Abdullah (2016A7PS0562H)
3) Deepak Gupta (2016A7PS0105H)
4) Mridul Bhaskar (2016A7PS0391H)

Dataset

Dataset was chosen from Movielens, with about 6000 users and 4000 movies. A ratings matrix was prepared and picked with each row representing a user and each column representing a movie.

Process of evaluating metrics for different types of recommendation techniques

1) Collaborative and collaborative baseline

   For collaborative and collaborative baseline approaches, a similar method was followed with minor extra steps for the baseline approach.

   Steps:

   1.1) Make a copy of the original ratings matrix.

   1.2) Zero down the values for the test-set region in the copy. The test-set region is the upper left corner of the matrix, with one-third of total rows and one-third of total columns. So the ratio **train-test split is about 8:1**.

   1.3) The we mean normalize and magnitude normalize each movie vector. In our case they are the columns of the copy matrix.

   1.4) Then we find similarity matrix between movies, by taking the transpose of the copy matrix and multiplying it with the copy matrix.

   1.5) **[Just for baseline]** We find the values of bi (for each user i), bj (for each movie j) and global_mean.

   1.6) Now we start parsing the test set region (only those values which are non-zero in the original ratings matrix)

   1.7) For each non-zero cell in the original ratings matrix, we make a prediction here.

   1.8) Say the cell is represented as (test_user, test_movie). We look for “k” closest movies to the test_movie from the training region**. (k=50 in our case)**

   1.9) We take the weighted average of those ratings on the basis of similarity scores and that is our prediction for the same cell.

   1.10) **[Just for baseline]** We include local and global effects in the predicted value also. That way we take care of **strict and lenient raters separately.**

1.11) After all test cells have been parsed, we find **RMSE** and **Spearman Rank** by standard methods.
RMSE = sqrt((sum of squares of errors over all cells)/(number of test cells))
Pearson Rank = 1-6*(sum of squares of errors over all cells)/(number of test cells^3 - number of test cells)
1.12) For **Precision on top K,** we consider every user in the test-set.
1.13) For each user in test-set, we find the top (K or number of relevant movies, whichever is lesser - relevance defined below) according to the predicted ratings.
1.14) Then we check how many movies out of those K movies are **relevant**. (How to decide relevance ? **We assumed a threshold of 3.** Any original rating greater than equal to 3 will be relevant and anything less is non-relevant. This is a "binarization" of the ratings.
1.15) Finally we average out the precision for the users.

2) SVD
For both SVD and SVD with 90% energy the **assumptions** are more important. The assumptions made are as follows :
   a) **RMSE** = sqrt((sum of squares of reconstruction errors over all cells)/(number of cells))
   b) **Pearson Rank** = 1-6*(sum of squares of reconstruction errors over all cells)/(number of cells^3 - number of cells)
   c) Precision on top K = We do the same thing as the for collaborative case, just this time, it's for all the users.

Steps:
   2.1) Use the built SVD routine to split the ratings matrix.
   2.2) For normal SVD, retain all eigenvalues and for 90% SVD, retain lesser eigenvalues, enough to preserve just 90% energy.
   2.3) Calculate reconstruction errors and evaluate metrics as above.

3) CUR
Assumptions and steps are broadly similar to the SVD case.

4) SVD_Collab
Steps:
   4.1) Calculate SVD of ratings_matrix
   4.2) Map the ratings_matrix into smaller concept space
   4.3) movie_to_concept = ((ratings_matrix)' * (U))'
   4.4) Magnitude normalize the movie_to_concept matrix along the columns
   4.5) similarity_matrix=(movie_to_concept)' * (movie_to_concept)
   4.6) Proceed as in collaborative

5) CUR_Collab

Steps:

5.1) Calculate CUR of ratings_matrix

5.2) Map the ratings_matrix into smaller concept space

5.3) movie_to_concept = ((ratings_matrix)' * (C))'

5.4) Magnitude normalize the movie_to_concept matrix along the columns

5.5) similarity_matrix=(movie_to_concept)' * (movie_to_concept)

5.6) Proceed as in collaborative

Packages Used

1) numpy = for matrix multiplication, eigenvalue decomposition, mean calculations, transpose finding, diagonalization of a vector into a square matrix etc.
2) pickle = to pickle and unpickle data structures
3) math = for basic square root purposes
4) time = for calculating running times

Results

	RMSE	**Precision on top K (K=20)**	**Spearman Rank Correlation**	**Time Taken**
Collaborative	**1.318**	**0.796**	**0.999**	**105.52**
Collaborative with baseline	**1.536**	**0.7508**	**0.999**	**529.48**
SVD	**0**	**1**	**1.0**	**296.92**
SVD with 90% energy	**0.0043**	**0.99**	**1.0**	**292.81**
CUR	**0.7718**	**0.993**	**1.0**	**873.14**
CUR with 90% energy	**0.767**	**1.0**	**1.0**	**832.31**
SVD_Collab	**0.9357**	**0.791**	**0.999**	**446.305**
CUR_Collab	**0.938**	**0.7948**	**0.999**	**1023.12**

1) SVD and SVD_90 time is inclusive of both SVD and the reconstruction to find RMSE.
2) Same for CUR and CUR_90.
3) Collaborative and Collaborative baseline time are inclusive of finding similarity matrix and calculating RMSE.
4) SVD_Collab and CUR_Collab include the matrix factorization, prediction and calculation of all metrics (not just the RMSE)