Documentation

collaborative_new.py
collaborative_baseline.py

mean_normalize_movies (matrix): mean normalizes an input matrix along the columns
magnitude_normalize_movies (matrix) : magnitude normalizes an input matrix along the columns
test_train_split(matrix) : makes a portion of top-left corner to be zeros. This becomes the train set.
ratings_matrix : Original matrix
temp_ratings: Copy of the ratings_matrix which mean and magnitude normalized
train_matrix: Alias of temp_ratings
similarity_matrix: Movie-movie similarity matrix
predicted_preferred: List of lists. For each user a list of tuples is stored. One tuple is the (predicted_rating, movie)
preferred_movies: List of tuples (as defined earlier) for each user in an iteration
sum_of_scores: Sum of Precision on Top K for all test users
squared_error: Sum of squared errors
number_of_test: Number of valid test set cells
neighbours: Number of neighbours to be considered
global_mean:global mean rating
b_users: Biases for users
b_movies: Biases for movies

SVD.py

mean_normalize_movies (matrix): mean normalizes an input matrix along the columns
magnitude_normalize_movies (matrix) : magnitude normalizes an input matrix along the columns
test_train_split(matrix) : makes a portion of top-left corner to be zeros. This becomes the train set.
ratings_matrix : Original matrix
debug: For debugging purposes
energy_fraction(s,frac): Returns number of eigenvalues to be retained to retain a frac of total energy in a singular value matrix s
SVD(matrix): Returns the SVD of the matrix
reconstructed: Reconstructed matrix

CUR.py

a: Original matrix
pr: Probability Matrix for rows
cr: Probability Matrix for columns
c: Number of rows/columns to be chosen
rows: R matrix after decomposition
cols: C matrix after decomposition
u: U matrix after decomposition
cur: Reconstructed Matrix

SVD_collab.py

mean_normalize_movies (matrix): mean normalizes an input matrix along the columns
magnitude_normalize_movies (matrix) : magnitude normalizes an input matrix along the columns
test_train_split(matrix) : makes a portion of top-left corner to be zeros. This becomes the train set.
ratings_matrix : Original matrix
debug: For debugging purposes
energy_fraction(s,frac): Returns number of eigenvalues to be retained to retain a frac of total energy in a singular value matrix s
SVD(matrix): Returns the SVD of the matrix
reconstructed: Reconstructed matrix
movie_to_concept: Movie to concept matrix**similarity_matrix**: Movie-movie similarity matrix
predicted_preferred: List of lists. For each user a list of tuples is stored. One tuple is the (predicted_rating, movie)
preferred_movies: List of tuples (as defined earlier) for each user in an iteration
sum_of_scores: Sum of Precision on Top K for all test users
squared_error: Sum of squared errors
number_of_test: Number of valid test set cells
neighbours: Number of neighbours to be considered

CUR_collab.py

a: Original matrix
pr: Probability Matrix for rows
cr: Probability Matrix for columns
c: Number of rows/columns to be chosen
rows: R matrix after decomposition
cols: C matrix after decomposition
u: U matrix after decomposition
cur: Reconstructed Matrix
movie_to_concept: Movie to concept
similarity_matrix: Movie-movie similarity matrix
predicted_preferred: List of lists. For each user a list of tuples is stored. One tuple is the (predicted_rating, movie)
preferred_movies: List of tuples (as defined earlier) for each user in an iteration
sum_of_scores: Sum of Precision on Top K for all test users
squared_error: Sum of squared errors
number_of_test: Number of valid test set cells
neighbours: Number of neighbours to be considered