TASK 3

Recommandations

Overview

A cute story.

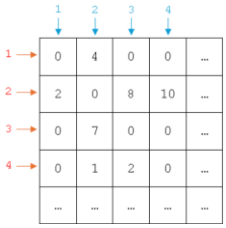
Story time

In a far away magical world, there was a fairy named Nixy. She was in the business of relaxation through dreaming and offered a wide range of themes for dreams to her clients.

One day, Captain Crook, a big client of Nixy, told her that he liked the previous dream a lot and wanted something similar, but different. Nixy has gathered a wide range of reviews from her clients and wants you to help her process the data to find the best dream for Captain Crook.

Data Structure

The reviews are strctured as a matrix. Each row represents a client and each column represents a theme. The value in the matrix is the rating of the client for the theme. The rating is an integer between 1 and 10.



Each cell (i,j) in the matrix represents the rating of client i*i* for theme j. It is 0 if the client did not rate the theme.

Preprocessing

Given the very high volume of data, we want to reduce the dimensionality of the matrix by eliminating the clients that contributed the least.

Task

The function preprocees takes in a matrix A and an integer min_count*min_count*. It removes from the matrix all clients (rows) that gave **strictly less** than min_count reviews.

Cosine Similarity

How similar are two themes?

The **cosine similarity** is a measure of similarity between two non-zero vectors. It is defined as the cosine of the angle between the two vectors. The cosine similarity is widely used in various applications, such as information retrieval, recommendation systems, and clustering.

Comparing two themes

In the context of recommendation systems, we can use the cosine similarity to compare two themes. Given two themes represented as vectors x*x* and y*y*, the cosine similarity is given by:

$$\text{cosine\_similarity}(x, y) = \frac{x \cdot y}{\|x\|_2 \|y\|_2}$$

where · denotes the dot product.

In our case, if the cosine similarity is close to 1, the two themes are similar as the angle between them is small. If the cosine similarity is close to -1, the two themes are dissimilar as the angle between them is close to 180 degrees (they point in opposite directions). If the cosine similarity is close to 0, the two themes are orthogonal.

Task

The function cosine_similarity takes in two column vectors, normalizes them, and computes the similarity using the above formula.

Singular Value Decomposition

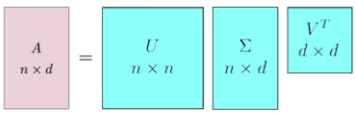
The **Singular Value Decomposition (SVD)** is a matrix factorization technique, based on singular vectors and singular values. Recall that the singular value equation for a matrix A*A* is given by:

$$Av = \sigma u$$
$$A^T u = \sigma v$$

where u*u* and v are the left and right singular vectors of A, respectively, and σ is the singular value. The SVD of a matrix A is given by:

$$A = U \Sigma V^T$$

where U and V are orthogonal matrices, and Σ is a diagonal matrix containing the singular values of A

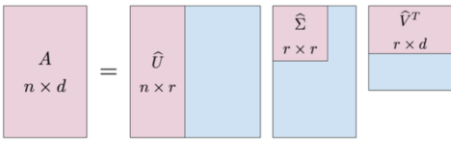


.

This decomposition exists for any complex matrix A, and it is possible to choose it such that the singular values are ordered in decreasing order. This is very important as each singular values represents the importance of the corresponding singular vectors in building the matrix A.

Reduced SVD

In practice, we often use a reduced version of the SVD. Given a matrix A of size n×d, the compact SVD is given by: $A = U_r \Sigma_r V_r^T$ where Ur is a matrix of size n×r, Σr is a diagonal matrix of size r×*d*, and Vr is a matrix of size r×d. The reduced SVD is obtained by keeping only the first r singular values and their corresponding singular vectors.



In our case, Ur contains the information about the clients and Vr*Vr* contains the information about the themes.

Task

The function recommendations takes in the path to the .csv file, the index of the liked theme, the number of themes to recommend, the number of minimum reviews and the number of singular values to keep. It returns the indices of the recommended themes.

The flow is as follows:

1. Load the matrix from the .csv file by implementing the read_mat function. (Hint: csvread)
2. Preprocess the matrix using the preprocess function.
3. Compute the reduced SVD of the matrix. (Hint: svds)
4. The V matrix will contain the information about the themes. Sort the themes by the similarity with the liked theme. Use the cosine_similarity function.
5. Return the indices of the recommended themes.