॥ सा विद्या या विमुक्तये ॥

भारतीय प्रौद्योगिकी संस्थान धारवाड

Indian Institute of Technology Dharwad

Mathematics for Data science

CS-427

ASSIGNMENT-1

GROUP MEMBERS

No.	NAME	ROLL NO.	BRANCH
1	Aduma Rishith Reddy	210010002	CSE
2	Gorantla Nikhil Sai	210010018	CSE
3	A.V.S.Sreenivasu	210020001	CSE
4	Anshif	210150004	EP

Contents

1 Singular Value Decompostion

The Singular Value Decomposition (SVD) of a matrix is a factorization of that matrix into three matrices. **The SVD of mxn matrix A is given by the formula :**

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

* U is mxn matrix of the orthonormal eigenvectors of AA^T.
* V^T transpose of a nxn matrix containing the orthonormal eigenvectors of A^TA.
* Σ is a nxn diagonal matrix of the singular values which are the square roots of the eigenvalues of A^TA/AA^T.

2 IMAGE COMPRESSION

* We can consider an image to be an mxn matrix, say A
* Using SVD,we can split the A into 3 matrices U,Σ,V^T.
  Here Σ is diagonal matrix whose diagonal entries are the singular values σ_1,σ_2... which act as scaling factors for the vectors of UV^T.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m,1} & u_{m,2} & \cdots & u_{m,r} \end{bmatrix} \begin{bmatrix} \sigma_{1,1} & 0 & \cdots & 0 \\ 0 & \sigma_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{r,r} \end{bmatrix} \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,r} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n,1} & v_{n,2} & \cdots & v_{n,r} \end{bmatrix}^T$$

* A coloured image consists of three channels R,G and B each of which is is an mxn matrix that consists of the color intensities ranging from 0 to 255.
* We will apply SVD on these channels individually and then combine them together to reconstruct the compressed image.
* If we apply SVD directly we are prone to get improper results so we normalize the data and then apply SVD. For normalization we do $x = (x - mean(x))/std(x)$ where x is a row of the image i.e we are trying to capture how different a pixel is to the remaining pixels. We also save these means and standard deviations so that we can get the final values in the $0 - 255$ range after compressing.
* We then used the svd() function from numpy.linalg module to compute the SVD of the normalized image channel array. This function gives us the singular values and the corresponding U, V^T in descending order of magnitude.

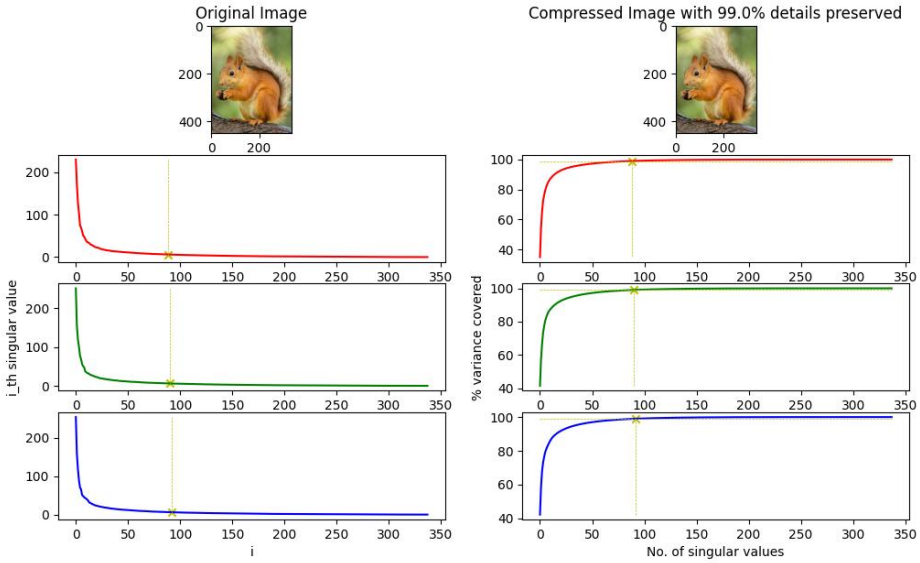
* We now want to reduce the number of singular values and the corresponding vectors from U, V that we are storing while preserving as much of the image as we can.

* The get_retain_dimensions() function in the ImageCompressor class gives the number of dimensions/singular values that we need to save to preserve some percent of the original image.

* We do this by considering the variance that is encapsulated by the singular values. Let's say that we want to compress our image preserving 99% of its details then we take the singular values till the sum of their squares is equal to 99% of the total sum of their squares.

* Now that we have the number of dimensions that we have to retain for a channel we discard the rest and reconstruct the image by multiplying these truncated U, S and V^T and then reverting them back to the $0 - 255$ range.

* Consider the following image of a squirrel. From the graphs we can see that the initial singular values have a higher magnitude than the later ones so most of the data is stored in them. We can see that we have to use roughly half of singular values from each channel to retain 99% of the original image.



* The compression precentage/ratio for this squirrel image is roughly 2.14. So we are preserving 99% of the image using only less than 40% of the space.

* We observed that if we reduce the retain percentage further though the space required drastically decreases it also affects the quality a lot.

Video Compression

* A Video is nothing but a sequence of images so we can extract individual frames from a video and apply SVD on each one of them and then recombine them into a video again.

* For this assignment we used a 17s long video which we were able to compress from 658KB to 336KB.

* The code for this may give an error depending on the pc. This is due to the different fourcc codes supported the system, If the code doesn't run on your pc please try with different codes like h264,x264,... the size of the compressed video also depends on the code used.

3 Movie Recommendations

- We have combined the ratings and movie csv files from the supplied data into a matrix, where each row corresponds to a user and each column to a movie. We then normalized the ratings matrix.
- We then applied SVD and received

$$U, \Sigma, V^T$$

- In this resulting decomposition we know the rows of U are users and columns of V^T are movies.
- However, we are unable to determine what each of these rows and columns mean. They might be connected to certain genres or another relationship that we are not aware of. However, this does not prevent us from using these developed features to compare two users' characteristics.
- The cosine similarity function, which assesses similarity between two vectors in an inner product space, is used to identify similarities. It establishes whether two vectors are roughly pointing in the same direction(lower the angle,higher the similarity) by calculating the cosine of the angle between them.

$$S_C(A, B) = cos(\theta) = \frac{A.B}{\|A\|\|B\|}$$

- We use truncated SVD to reduce the dimension of matrix U. In essence, this means we are removing several columns on U that the corresponding singular values in s that are small,i.e. They have less significance in deciding what to recommend, before we use it to compute the similarity.
- Using the above concepts we found the most similar group of people to the target person (person we would like to recommend movies to) and recommend the movies which are rated highly by the reference person and haven't been watched by our target person.
- Our program recommends movies for the user whose userid can be specified in the code block 9. If you need recommendations for a new user, please append their ratings to the 2-d array ratings and change userid to the respective id and run the entire process again to obtain those recommendations.