Full Resolution Image Compression using SVD, PCA and K means Clustering: A Comparative Analysis

By- *Srishti Singh (srishti.singh1996@gmail.com)*
Parinita Bohra (bohraparinita123@gmail.com)
IISc Computing in AI and ML Project Report

Introduction

As the technology is becoming an important part of day-to-day life, large volumes of data is being generated on a daily basis. This data can be of various forms, such as text, audio, video, images and so on. Images are one of the most popular data sharing method. Since 'A picture is worth a thousand words', we often exchange images to convey information. Revolution in the field of smart phones and smart devices has enabled the uses to use the pictorial data more conveniently. This way, we data scientists store, process, transmit huge amount of digital data on a daily basis. Storing the uncompressed data prove to be very expensive in terms of storage cost. Also, transmission of uncompressed data requires large bandwidth. Because of these reasons, techniques of compressing data before storage and transmission has become a significant area of interest in the research community; particularly in the fields such as artificial intelligence, pattern recognition, signal processing and so on. In this paper, we focus on reducing the storage space needed for storing an image file using the technique of singular value decomposition. Image compression deals with reducing the size of data required to represent images. As the images are digital, goal of the image compression algorithm is to represent the image with lowest number of bits possible. Redundancies present in the image file can be exploited to in order to reduce the number of bits. Redundancy can be defined as approximate repetitive pattern that tend to give the degree of resolution to the image. However, quality of the image should not be compromised too much so that it becomes incomprehensible for the user. A good image compression algorithm should strike a balance between this trade-off. Usually the requirement of the application dictate the trade- off between compression and quality of the data. Image compression techniques can be broadly classified into two categories. Lossless compression and lossy compression. Lossless compression techniques defines entropy, which limits the reduction in the image. It tends to produce the same copy as that of the original data. Lossless compression is known as reversible as it doesn't degrade the quality of the image. Lossy compression techniques identify the minute details and variations in the system which the human eye is not fine-tuned to recognize. Amount of storage required to store the image file can be reduced by eliminating such features. Lossy compression often compromises the quality of the image. However, it can be used to achieve the storage space requirements. Lossy compression is irreversible because it degrades the data. Compression possible in lossy techniques is much higher than lossless techniques . Image is generally represented using a matrix, which is of size m×n. Here m is the number of rows, which signifies the pixel height of the image and n is the number of columns which signifies the pixel width of the image. Each element of the matrix gives the representation to each pixel that make up the image. In an image, darkness or brightness of a pixel relative to other pixels is represented by assigning a number to each pixel. In this way, each element

of the matrix decides the characteristics of the corresponding pixel. In case of black and white images, each value in the matrix ranges between 0 (black) and 1 (one), which translates into the relative grayness of each pixel. Colored images are composed by three basic colors; red, green and blue (RGB). Every time a colored image is being stored, it is split by the computer into these three layers. Because of the additional layer involved, colored images consume more space when compared to grayscale images. So for a colored image, each pixel can be associated with three values, ranging from 0 (not colored) to 1 (saturated). These three values are assigned to red, green and blue. In this project we compare three algorithms for image compression using compression ratio as their metric of performance.

Methodology

SVD

In singular value decomposition method, digital image is given to SVD. SVD refactors the given digital image into three matrices. Singular values are used to refactor the image and at the end of this process, image is represented with smaller set of values, hence reducing the storage space required by the image. Goal here is to achieve the image compression while preserving the important features which describe the original image. SVD can be adapted to any arbitrary, square, reversible and non-reversible matrix of m × n size. Singular Value Decomposition (SVD) generally works by decomposing the original matrix. SVD aims to approximate the dataset of large number of dimensions using fewer dimensions. SVD considers a highly variable, high dimensional data points and exposes the substructure of the original data by reducing the higher dimensional data into lower dimensional data. Exposure of the substructure orders the data from most variation to the least. This helps to find the region of most variation and then later SVD can be used for reduction. SVD is basically factorizing the given matrix into multiple matrices. SVD factorizes the given matrix with m number of rows and n number of columns it into three matrices; which can be written as $M = U\sum V^T$ where U and V^T are orthogonal matrices of the order m×m and n×n respectively and $\sum$ is a diagonal matrix of the order m×n. Diagonal matrix $\sum$ consists of nonnegative real numbers which are called as singular values of M. The m columns of U and n columns of V^T are known as left and right singular vectors of M respectively . SVD of a given matrix can be calculated as follows: • From a given matrix M, calculate AA^T and A^TA. • Use AA^T to form U, which is calculated by calculating eigen values and eigen vectors of AA^T . • In the same way, V can be formed by calculating the eigen values and eigen vectors of A^TA. • Columns of U and V are formed by dividing each eigen vector by its magnitude. • Singular values are computed by taking the square root of eigen values. They are arranged in descending order in the diagonal matrix.

PCA

The principal component analysis (PCA) allows to analyse and visualize a dataset containing individuals described by several quantitative variables. It is a statistical method which makes it possible to explore so-called multivariate data (data with several variables). Each variable could be considered as a different dimension. If you have more than 3 variables in your dataset, it could be very difficult to visualize the data in a multidimensional "hyperspace". Principal component analysis is used to extract and visualize the important information contained in a multivariate data table. The PCA synthesizes this information into just a few new variables called principal components. These

new variables correspond to a linear combination of the original variables. The number of main components is less than or equal to the number of original variables. The information contained in a data set corresponds to the variance or the total inertia it contains. The objective of the PCA is to identify the directions (i.e., main axes or main components) along which the variation in the data is maximum. In other words, PCA reduces the dimensions of multivariate data to two or three main components, which can be viewed graphically, losing as little information as possible. PCA is actually part of a set of data analysis methods, called multifactorial methods. In general, they aim to summarize as faithfully as possible a large set of data, that is to say different observations (the variables) for each member of a large population of study (individuals). This summarization always causes a loss of information, but it is for the benefit of the most relevant information and readability, therefore of the best interpretation.
There are several different approaches to PCA, but all agree on the conditions for its application and its general objective. This method applies to quantitative data sets of at least two variables. Since this is a multifactorial data analysis method, its purpose is to summarize this dataset. This is done
in the following ways: Primarily, Provide simple and readable tools for representing the information processed, making it possible to bring out from the raw data the possible links existing between the variables (in terms of correlation)
Secondary, give indications on the nature, the strength and the relevance of these links, in order to facilitate their interpretation and discover which the dominant trends of the dataset are.
Thirdly, effectively reduce the number of dimensions studied (and thus simplify the analysis), by seeking to express the original set of data as faithfully as possible thanks to the relationships detected between the variables.

Advantages of PCA

Mathematical simplicity: PCA is a factorial method because the reduction in the number of characters is not done by a simple selection of some of them, but by the construction of new synthetic characters obtained by combining the initial characters using "factors". However, these are only linear combinations. The only real mathematical tools used in PCA are the calculation of the eigenvalues /vectors of a matrix, and the basic changes. Mathematically, PCA is therefore a simple method to implement.
• Simplicity of results: Thanks to the graphs it provides, the Principal Component Analysis makes it possible to grasp a large part of its results at a glance.
• Power: Although PCA is simple, it is no less powerful. It offers, in a few operations only, a summarization and a complete view of the relationships existing between the quantitative variables of a studied population. These results could not have been obtained otherwise, only at the cost of tedious manipulations.
• Flexibility: PCA is a very flexible method, since it is applied to a set of data of any content and size, as long as it is quantitative data organized in the form of individuals/variables. This flexibility of use is mainly reflected in the diversity of PCA applications, which affects all areas.

Disadvantages of PCA

As a method of data analysis, the PCA does not really have major disadvantages. It would therefore make no sense to say that it is a drawback of the PCA that it does not apply outside this context. Likewise, given that it is above all a technique of summarizing data, the inevitable loss of information is not a drawback, but rather a condition for obtaining the result, even if it obscures sometimes characteristics which are nevertheless representative in certain particular cases.

K Means Clustering

k-means clustering is unsupervised learning algorithm. It is a method of vector quantization, that aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean (cluster centres or cluster centroid), serving as a prototype of the cluster. This results

in a partitioning of the data space into Voronoi cells. *k*-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

The problem is computationally difficult (NP-hard); however, efficient heuristic algorithms converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both *k-means* and *Gaussian mixture modeling*. They both use cluster centers to model the data; however, *k*-means clustering tends to find clusters of comparable spatial extent, while the Gaussian mixture model allows clusters to have different shapes

We can understand the working of K-Means clustering algorithm with the help of following steps −

- Step 1 − First, we need to specify the number of clusters, K, need to be generated by this algorithm.
- Step 2 − Next, randomly select K data points and assign each data point to a cluster. In simple words, classify the data based on the number of data points.
- Step 3 − Now it will compute the cluster centroids.
- Step 4 − Next, keep iterating the following until we find optimal centroid which is the assignment of data points to the clusters that are not changing any more

First, the sum of squared distance between data points and centroids would be computed.

Now, we have to assign each data point to the cluster that is closer than other cluster (centroid).

At last compute the centroids for the clusters by taking the average of all data points of that cluster.

K-means follows Expectation-Maximization approach to solve the problem. The Expectation-step is used for assigning the data points to the closest cluster and the Maximization-step is used for computing the centroid of each cluster.

While working with K-means algorithm we need to take care of the following things −

- While working with clustering algorithms including K-Means, it is recommended to standardize the data because such algorithms use distance-based measurement to determine the similarity between data points.
- Due to the iterative nature of K-Means and random initialization of centroids, K-Means may stick in a local optimum and may not converge to global optimum. That is why it is recommended to use different initializations of centroids.

Result

Original Image (970KB)

Compressed Image (81 KB) obtained using K means of Clustering

The image that we use in our project for the compression is of Cosmic Tarantula captured by NASA's James Webb Space Telescope. We compared three algorithms namely SVD, PCA and K Means of clustering for the purpose of image compression. The metric used for this comparison is Compression Ratio (CR)

CR = size(Compressed) / size(Image) * 100

This project was carried in python programming language using several of its libraries. As per our observation the CR obtained by the K means of Clustering was 8.35 as the space occupied by the compressed file is 81 kb while with PCA it is 10.467 and even higher with SVD. Thus it is seen that with compressed image obtained by the K means algorithm occupies less storage space as compared to other algorithms. It is capable of retaining the quality of image even after compression.

References

[1] C. -W. Wang and J. -H. Jeng, "Image compression using PCA with clustering," *2012 International Symposium on Intelligent Signal Processing and Communications Systems*, 2012, pp. 458-462, doi: 10.1109/ISPACS.2012.6473533.

[2] J. McNeely and G. Geiger, "K-Means Based Spatial Aggregation for Hyperspectral Compression," *2014 Data Compression Conference*, 2014, pp. 416-416, doi: 10.1109/DCC.2014.15.

[3] M. Wei, Y. Zhao, J. Lu, X. Chen, C. Li and Z. Li, "Convolutional Neural Network Accelerator for Compression Based on Simon k-means," *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1-7, doi: 10.1109/IJCNN55064.2022.9892097

[4] B. A. Lungisani, C. K. Lebekwe, A. M. Zungeru and A. Yahya, "Image Compression Techniques in Wireless Sensor Networks: A Survey and Comparison," in *IEEE Access*, vol. 10, pp. 82511-82530, 2022, doi: 10.1109/ACCESS.2022.3195891.

[5] X. Jin and Z. Sha, "Facial Expression Recognition Based on K-SVD Dictionary Learning and OMP Compression Perceptual Reconstruction," *2021 8th International Conference on Computational Science/Intelligence and Applied Informatics (CSII)*, 2021, pp. 49-54, doi: 10.1109/CSII54342.2021.00018.