Principal Component Analysis

VIT
Vellore Institute of Technology

Soumyadeep Ganguly (24MDT0082)

MSc Data Science,

PMDS501L: Linear Algebra

Prof. Rushi Kumar B

November 10, 2024

Abstract

The principles and uses of Principal Component Analysis (PCA) as a dimensionality reduction method for high-dimensional datasets in data science are examined in this research. By projecting data onto new coordinates based on the axes of highest variation, PCA is crucial for converting complex data into a more interpretable format. The mathematical underpinnings of PCA, such as eigenvalues and eigenvectors, as well as its applications, such as data normalization, are covered in detail in this study. A detailed implementation process for PCA is given, and then an example application in picture compression is shown. The outcomes show how well PCA reduces data size while maintaining important information, which makes it useful for image processing, anomaly detection, and machine learning.

Keywords: PCA, Dimensionality Reduction, Image Processing

Introduction

Successfully organizing and interpreting enormous amounts of data has become crucial to contemporary data science in the age of big data and high-dimensional datasets. The difficulty of deriving valuable insights grows with the amount and complexity of data. Because high-dimensional data frequently contains duplicate or redundant components that could obscure critical correlations, it can be challenging to visualize and comprehend. Dimensionality reduction has become a crucial data pre-processing approach to address these issues. The goal of dimensionality reduction is to minimize the quantity of input variables in a dataset while retaining the greatest amount of pertinent information. It improves interpretability, speeds up computations, and uses less storage by making the structure of the data easier.

Principal Component Analysis (PCA) is one of the most renowned and often applied methods to achieve dimensionality reduction. PCA, which has its roots in linear algebra, provides an organized approach to converting complicated, multi-dimensional data into a more straightforward and manageable structure. Projecting data onto a new coordinate system whose axes, or principle components, match the directions of the data's greatest variance is the fundamental idea of principal component analysis (PCA). While these major components are orthogonal, each one is guaranteed to collect distinct information without overlapping. Underlying patterns in the data that might not be visible in its original high-dimensional space are made apparent by this modification.

Beyond just simplifying data, PCA can also efficiently counteract the "curse of dimensionality," a phenomenon in which data gets sparse as the number of dimensions' rises, resulting in subpar model performance and an increased computing load. PCA guarantees that the dataset stays useful while minimizing noise and redundancy by identifying the most important components of the data and eliminating those that offer little to the variance. This method has significant ramifications for machine learning, allowing for better feature selection, increased model truthfulness, and more effective algorithm training.

The article explores the mathematical foundations of principle component analysis (PCA), looking at how ideas from linear algebra, such as eigenvectors and eigenvalues, are essential for identifying principal components. Practical issues are also addressed, including the significance of standardization data prior to PCA application and how to evaluate the altered dataset that follows. Readers will grasp why PCA is still a vital tool for data analysis and pre-processing in a variety of scientific and industrial fields by developing an in-depth comprehension of its physics and applications.

Literature Review

PCA originated from Pearson's early 20th-century work and was formalized by Hotelling in 1933. Abdi and Williams note that PCA relies on Eigen-decomposition and SVD to transform data into principal components, linear combinations that maximize data variance. This approach simplifies complex multivariate data, making it easier to detect patterns and relationships (Abdi, 2010).

Greenacre et al. emphasize the importance of data standardization to avoid skewed results. PCA uses eigenvalue decomposition and SVD to reduce data dimensionality by extracting principal components. These components are then used to create biplots that visually map data points and variables (reenacre, 2022).

PCA extends beyond simple data reduction; it can handle qualitative data through correspondence analysis, as highlighted by Abdi and Williams. Greenacre et al. also describe applications in missing

data estimation and sparse component analysis for high-dimensional data. This adaptability showcases PCA's usefulness across various fields (Abdi, 2010) (reenacre, 2022).

When (Jamal, 2018) employed PCA and K-means clustering to predict breast cancer, they discovered that PCA was good at reducing dimensionality, while K-means, which isn't as often used for this, did similarly well, showing promise as a feature extraction substitute.

(Reddy, 2020) examines Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for dimensionality reduction, applied to various machine learning algorithms. Results indicate that PCA generally performs better, especially on high-dimensional datasets, improving classifier efficiency and accuracy. Decision Tree and Random Forest algorithms remain robust, while LDA often reduces performance, particularly in terms of specificity and accuracy. For smaller datasets, dimensionality reduction negatively affects results. Overall, PCA is recommended for large datasets, with Random Forest and SVM yielding the best outcomes.

With an emphasis on Support Vector Machine (SVM) for classification and Principal Component Analysis (PCA) for dimensionality reduction, the study investigates anomaly detection through machine learning. PCA efficiently reduces the feature set from 42 to 28 using the KDD99 dataset, increasing execution speed and improving classification performance by reducing misclassifications. The method is effective for network intrusion detection systems since PCA and SVM together show better precision and recall than SVM alone. This technique will be used in future research to identify new abnormalities and other kinds of data (George, 2012).

Methodology

High-dimensional data is transformed into a more manageable and interpretable form through a number of systematic processes in the Principal Component Analysis (PCA) implementation process. In order to capture variable associations, a covariance matrix is computed after data preprocessing to guarantee consistency. The most important components are found using eigenvalue and eigenvector decomposition, which allows a projection onto a reduced-dimensional space. Every stage simplifies the complexity of the data while guaranteeing the preservation of the greatest amount of variance. The steps involved in applying PCA for dimensionality reduction and further data analysis are described in detail below.

Step 1

Data Preprocessing:

Centre the data by subtracting the mean of each variable to ensure a zero-mean dataset. Standardize variables if they have different scales to make sure each contributes equally to the analysis.

Calculate Mean $\bar{x} = \frac{\Sigma_{k=1}^{N} x_k}{N}$

Step 2

Covariance Matrix Calculation:

Compute the covariance matrix of the cantered data to capture the relationships between variables.

Calculation of Covariance Matrix:

$$Cov(x_m, x_n) = \frac{1}{(N-1)} \sum_{k=1}^{N} (x_{mk} - \bar{x}_m)(x_{nk} - \bar{x}_n)$$

Covariance matrix is:

$$S = \begin{bmatrix} Cov(x_1, x_1) & \cdots & Cov(x_1, x_n) \\ \vdots & \ddots & \vdots \\ Cov(x_m, x_1) & \cdots & Cov(x_m, x_n) \end{bmatrix}$$

Step 3

Eigenvalue and Eigenvector Decomposition:

Decompose the covariance matrix to extract eigenvalues and their corresponding eigenvectors. Eigenvalues represent the variance explained by each component, while eigenvectors indicate the direction of the components.

Assume, $\lambda \; is \; Eigen \; Value,$

$$\begin{bmatrix} Cov(x_1, x_1) - \lambda & \cdots & Cov(x_1, x_n) \\ \vdots & \ddots & \vdots \\ Cov(x_m, x_1) & \cdots & Cov(x_m, x_n) - \lambda \end{bmatrix} = 0$$

Eigen Vector, $U = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$, $(S - \lambda I)U = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Normalized Eigen vector,

$$e_n = \begin{bmatrix} u_1 / \|U\| \\ \vdots \\ u_m / \|U\| \end{bmatrix}$$

Step 4

Selecting Principal Components:

Rank eigenvectors based on their eigenvalues (in descending order). Choose the top k eigenvectors corresponding to the largest eigenvalues to form the new feature space.

Principal Component,

$$P_n = e_n^T . \begin{bmatrix} x_{1k} - \bar{x}_1 \\ \vdots \\ x_{mk} - \bar{x}_m \end{bmatrix}$$

All principal components, $[P_1, P_{2,} \; \ldots, P_n]$

Step 5

Projection onto Principal Components:

Transform the original dataset by projecting it onto the selected principal components, yielding a reduced-dimensional representation.

Application: Image Compression

Principal Component Analysis (PCA) has useful applications in fields like picture compression in addition to being a powerful method for dimensionality reduction. High-dimensional picture data can be reduced to a smaller collection of principle components that best represent the variance by using PCA (Mofarreh-Bonab, 2012). By using fewer data points to describe image files while maintaining key visual characteristics, this technique enables image file compression. As a result, PCA-based image compression is useful for effective storage and transmission in a variety of applications since it may drastically reduce file sizes without sacrificing quality.

Sample Image

Fig 1: Sample image

$$\begin{bmatrix} 24 & \cdots & 125 \\ \vdots & \ddots & \vdots \\ 230 & \cdots & 23 \end{bmatrix}$$

Matrix form of the sample image

Three colour channels make up the RGB (Red, Green, Blue) sample image that was previously discussed. A (640 x 336) matrix is used to represent each channel, and each matrix, where each element is ranged between 0 to 255, represents the image's level of red, green, or blue intensity. The original coloured image is created by combining these three matrices. Because the final colour of each pixel is decided by the combination of the corresponding values from the three matrices, this structure enables a rich representation of colours. The foundation of colour modification and image processing methods, this multi-channel format is crucial for maintaining fine-grained colour information in digital images.

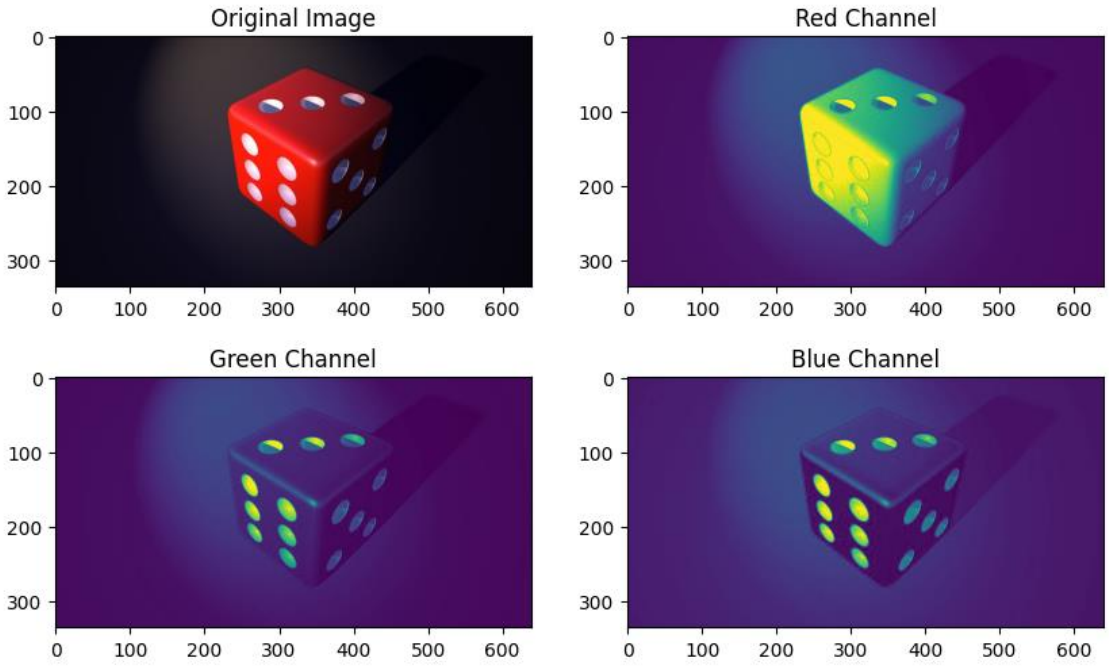

Fig 2: Original image and its multiple coloured channel

Each colour channel (Red, Green, and Blue) must be treated independently in order to perform PCA for image compression on an RGB image. Each channel undergoes PCA to lower its dimensionality while keeping the majority of the visual information in the image. Each colour channel matrix is flattened, principle components are extracted using PCA, and the image is then rebuilt using just these components. Because the rebuilt image will be much smaller, it may be transmitted and stored more effectively while still being of a respectable quality. Below is a step-by-step guide on how to achieve this.

Step 1: Load and Pre-process the Image

Load the image using an image processing library and split it into its three colour channels (Red, Green, and Blue).

Step 2: Identifying number of Principal Component

Finding the "elbow point," or the point at which the explained variance begins to level off, is necessary to determine the ideal number of principle components using a scree plot. Since adding more components only slightly helps to explain greater variance, this point identifies the most important components to keep.

Step 3: Apply PCA to Each Channel

Flatten each 640×336 matrix into a 2D format suitable for PCA.

Fit PCA to each channel, specifying the number of components to retain the desired variance level.

Step 4: Transform and Compress the Channels

Project each colour channel onto its principal components to reduce dimensionality.

Reconstruct the image channels from the reduced data.

Step 5: Combine Reconstructed Channels

Merge the reconstructed Red, Green, and Blue channels to form the final compressed RGB image.

Step 6: Visualize the Original and Compressed Images

Display or save the compressed image and compare it to the original to evaluate compression quality.

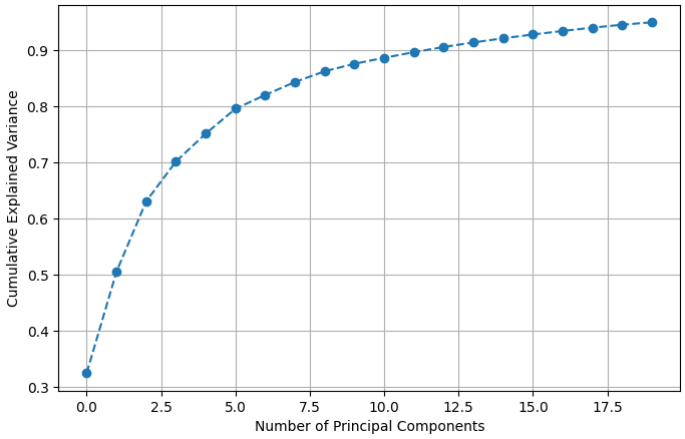

Fig 3: Scree plot for explained variance to identify optimal number of principal component.

Python Code

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
from sklearn.decomposition import PCA

def compress_image(file_name):
    image = cv2.cvtColor(cv2.imread(file_name), cv2.COLOR_BGR2RGB)

    red, green, blue = cv2.split(image)
    red, green, blue = red/255, green/255, blue/255

    pca_comp = PCA(n_components=20)

    reduced_red = pca_comp.fit_transform(red)
    reconstucted_red = pca_comp.inverse_transform(reduced_red)

    reduced_green = pca_comp.fit_transform(green)
    reconstucted_green = pca_comp.inverse_transform(reduced_green)

    reduced_blue = pca_comp.fit_transform(blue)
    reconstucted_blue = pca_comp.inverse_transform(reduced_blue)
 reduced_image=
cv2.merge((reconstucted_red,reconstucted_green,reconstucted_blue))
    cv2.imwrite("compresed_image.jpg", reduced_image*255)
    return "Image Compressed successfully"

if __name__ == "__main__":
    compress_image("image.jpg")
```

Result Analysis

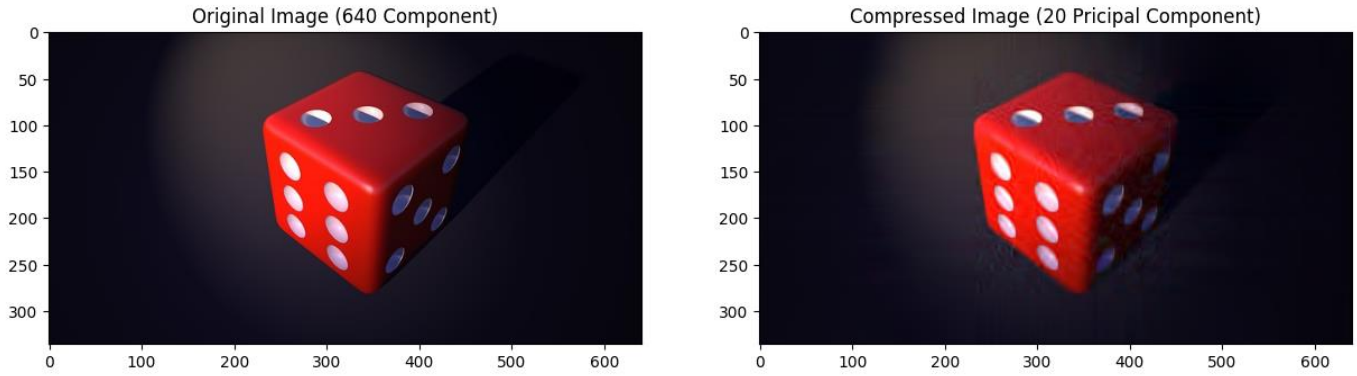

Fig 4: Result after applying PCA

Our method produced a compressed image with **twenty primary principal components**. We were able to drastically reduce the image's file size by lowering its dimensionality. In particular, the original image's size was decreased from 28 KB to 17 KB. The effectiveness of image storage in memory is improved by this size reduction, which makes it more appropriate for applications that demand resource management and optimal performance. Furthermore, dimensionality reduction ensures a balance between compression and image quality by preserving the image's key characteristics while reducing information loss.

Conclusion

It has been demonstrated that Principal Component Analysis (PCA) is a priceless method for dimensionality reduction, streamlining intricate datasets while preserving crucial information. PCA overcomes the difficulties presented by high-dimensional data by improving computing efficiency and model performance by lowering noise and redundancy. Finding the principle components that best capture data variation requires an understanding of the mathematical foundations of PCA, such as eigenvalue decomposition. Furthermore, the use of PCA for image compression shows how useful it is in maintaining image quality while using less storage space. PCA's significance in data science and analytics is highlighted by its versatility across domains, including machine learning, anomaly detection, and data visualization. In increasingly complex data contexts, PCA will remain a vital technique for improving data interpretation, efficiency, and insight finding, according to this paper.

References

Abdi, H. W. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 433--459.

George, A. a. (2012). Anomaly detection based on machine learning: dimensionality reduction using PCA and classification using SVM. *International Journal of Computer Applications*, 5--8.

Jamal, A. a. (2018). Dimensionality reduction using pca and k-means clustering for breast cancer prediction. *Lontar Komput. J. Ilm. Teknol. Inf*, 192--201.

Mofarreh-Bonab, M. a.-B. (2012). A new technique for image compression using PCA. *International Journal of Computer Science \& Communication Networks*, 111--116.

Reddy, G. T. (2020). Analysis of dimensionality reduction techniques on big data. *Ieee Access*, 54776--54788.

reenacre, M. a. (2022). Principal component analysis. *Nature Reviews Methods Primers*, 100.