Principal Component Analysis (PCA)

Introduction

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of a dataset while retaining most of the variation present in the data. It achieves this by transforming the data into a new set of orthogonal axes, called principal components, which are ordered by the amount of variance they capture from the data. PCA is widely used in data science for tasks such as feature reduction, data visualization, noise reduction, and improving the efficiency of machine learning algorithms.

Concept and Purpose

The primary goal of PCA is to identify the directions (principal components) in which the data varies the most and to project the data onto these directions. This helps in reducing the number of dimensions without losing significant information.

1. **Variance and Covariance**: PCA relies on the variance of the data and the covariance between different features to identify the principal components.
2. **Orthogonal Transformation**: The transformation involved in PCA is orthogonal, meaning the principal components are orthogonal (perpendicular) to each other.
3. **Dimensionality Reduction**: By selecting a subset of the principal components, PCA reduces the number of dimensions while preserving as much variability as possible.

Step-by-Step Procedure

1. **Standardize the Data**:
   - Center the data by subtracting the mean of each feature.
   - Scale the data by dividing by the standard deviation of each feature.
   - This ensures that each feature contributes equally to the analysis.

$$\mathbf{Z} = \frac{\mathbf{X} - \mu}{\sigma}$$

2. **Compute the Covariance Matrix**:
   - The covariance matrix captures the relationships between the different features.
   - For a standardized dataset $\mathbf{Z}$, the covariance matrix $\mathbf{C}$ is computed as:

$$\mathbf{C} = \frac{1}{n-1}\mathbf{Z}^T\mathbf{Z}$$

3. **Compute the Eigenvalues and Eigenvectors**:

- Eigenvalues and eigenvectors of the covariance matrix are computed.
- The eigenvectors represent the directions of maximum variance (principal components), and the eigenvalues indicate the magnitude of the variance in these directions.

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$$

4. **Sort Eigenvalues and Eigenvectors**:
   - The eigenvalues are sorted in descending order.
   - The corresponding eigenvectors are also sorted to align with the sorted eigenvalues.
5. **Select Principal Components**:
   - Select the top k eigenvectors corresponding to the largest k eigenvalues.
   - These selected eigenvectors form the principal components.
6. **Transform the Data**:
   - Project the original data onto the selected principal components to obtain the reduced-dimensional dataset.

$$\mathbf{Y} = \mathbf{Z}\mathbf{W}$$

Where $\mathbf{W}$ is the matrix of the selected eigenvectors.

Detailed Example

Consider a dataset with two features for simplicity. The dataset consists of 10 data points:

$$\mathbf{X} = \begin{bmatrix} 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \\ 2.3 & 2.7 \\ 2 & 1.6 \\ 1 & 1.1 \\ 1.5 & 1.6 \\ 1.1 & 0.9 \end{bmatrix}$$

1. **Standardize the Data**:

$$\mu = \begin{bmatrix} 1.81 \\ 1.91 \end{bmatrix}, \quad \sigma = \begin{bmatrix} 0.83 \\ 0.83 \end{bmatrix}$$

$$\mathbf{Z} = \frac{\mathbf{X} - \mu^T}{\sigma^T} = \begin{bmatrix} 0.82 & 0.61 \\ -1.57 & -1.45 \\ 0.46 & 1.20 \\ 0.10 & 0.35 \\ 1.55 & 1.32 \\ 0.58 & 0.94 \\ 0.23 & -0.31 \\ -0.97 & -0.97 \\ -0.37 & -0.37 \\ -0.85 & -1.22 \end{bmatrix}$$

2. **Compute the Covariance Matrix**:

$$\mathbf{C} = \frac{1}{9}\mathbf{Z}^T\mathbf{Z} = \begin{bmatrix} 1.12 & 1.09 \\ 1.09 & 1.15 \end{bmatrix}$$

3. **Compute the Eigenvalues and Eigenvectors**:

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$$

Solving for eigenvalues:

$$\det(\mathbf{C} - \lambda\mathbf{I}) = 0$$

$$\begin{vmatrix} 1.12 - \lambda & 1.09 \\ 1.09 & 1.15 - \lambda \end{vmatrix} = 0$$

The eigenvalues are $\lambda_1 = 2.02$ and $\lambda_2 = 0.25$.
The corresponding eigenvectors are:

$$\mathbf{v}_1 = \begin{bmatrix} 0.68 \\ 0.73 \end{bmatrix}, \quad \mathbf{v}_2 = \begin{bmatrix} -0.73 \\ 0.68 \end{bmatrix}$$

4. **Sort Eigenvalues and Eigenvectors**:
   - Eigenvalues are already sorted: $\lambda_1 = 2.02$, $\lambda_2 = 0.25$.
   - Eigenvectors corresponding to λ_1 and λ_2 are sorted.
5. **Select Principal Components**:
   - Select the eigenvector corresponding to λ_1 as the principal component since it captures the most variance.
6. **Transform the Data**:

- Project the standardized data onto the selected principal component:

$$\mathbf{Y} = \mathbf{Z}\mathbf{v}_1 = \begin{bmatrix} 0.82 & 0.61 \\ -1.57 & -1.45 \\ 0.46 & 1.20 \\ 0.10 & 0.35 \\ 1.55 & 1.32 \\ 0.58 & 0.94 \\ 0.23 & -0.31 \\ -0.97 & -0.97 \\ -0.37 & -0.37 \\ -0.85 & -1.22 \end{bmatrix} \begin{bmatrix} 0.68 \\ 0.73 \end{bmatrix}$$

The transformed data is:

$$\mathbf{Y} = \begin{bmatrix} 1.07 \\ -2.15 \\ 1.48 \\ 0.32 \\ 2.19 \\ 1.29 \\ -0.04 \\ -1.37 \\ -0.52 \\ -1.17 \end{bmatrix}$$

Applications in Dimensionality Reduction

1. **Visualization**: PCA is used to reduce high-dimensional data to 2 or 3 dimensions for visualization. This helps in understanding the structure and patterns in the data.
2. **Noise Reduction**: By keeping only the principal components with the highest variance, PCA can filter out noise and improve the quality of the data.
3. **Feature Extraction**: PCA transforms the data into a lower-dimensional space, which can be used as new features for machine learning models. This can lead to improved model performance and reduced computation time.
4. **Data Compression**: PCA can compress the data by reducing the number of dimensions, which is useful in applications like image compression and storage.