What is Dimensionality Reduction?

Dimensionality reduction is the process of reducing the number of input variables (features) in a dataset while retaining as much of the relevant information as possible. It transforms high-dimensional data into a lower-dimensional space.

Why is Dimensionality Reduction Needed?

1. **Curse of Dimensionality**:
    - As the number of dimensions increases, data becomes sparse, and models may struggle to find meaningful patterns.
    - High-dimensional data can lead to overfitting in machine learning models.
2. **Reduced Computational Complexity**:
    - Lower dimensions mean faster computations, making algorithms more efficient.
3. **Improved Visualization**:
    - Data in 2D or 3D is easier to visualize and interpret compared to higher dimensions.
4. **Noise Reduction**:
    - High-dimensional datasets often contain redundant or irrelevant features. Dimensionality reduction helps remove noise and focuses on the most important features.
5. **Storage and Memory Efficiency**:
    - Fewer dimensions reduce the storage requirements and memory usage, especially for large datasets.

How Does PCA Help in Dimensionality Reduction?

Principal Component Analysis (PCA) is one of the most widely used techniques for dimensionality reduction. Here's how it works:

1. **Identifies Important Features**:
    - PCA identifies the directions (principal components) in which the data varies the most. These directions capture the most significant patterns in the data.
2. **Transforms Data**:
    - PCA projects the data onto a new set of axes (principal components), where the first few components capture most of the variance.
3. **Retains Maximum Variance**:
    - By selecting only the top kkk principal components (those with the highest variance), PCA ensures that the reduced dataset retains as much of the original information as possible.

4. **Removes Redundancy**:
    - PCA removes correlations between features by transforming them into a set of uncorrelated components.
5. **Simplifies the Dataset**:
    - PCA reduces the number of features, simplifying the dataset while maintaining its essential structure.

Example: How PCA Reduces Dimensions

1. **High-Dimensional Data**:
    - Imagine a dataset with 10 features (dimensions). Not all features may be equally important; some may be correlated or irrelevant.
2. **PCA Process**:
    - PCA analyzes the data and identifies the top principal components. For example:
        - PC1: Explains 70% of the variance.
        - PC2: Explains 20% of the variance.
        - PC3: Explains 5% of the variance.
    - Together, PC1 and PC2 explain 90% of the variance.
3. **Dimensionality Reduction**:
    - Instead of using all 10 features, PCA reduces the dataset to just 2 dimensions (PC1 and PC2), retaining most of the original information.

Benefits of Using PCA for Dimensionality Reduction

- **Preserves Variance**: Ensures that the reduced dataset still captures the most important patterns.
- **Improves Model Performance**: By focusing on key features, PCA can improve the accuracy and efficiency of machine learning models.
- **Facilitates Visualization**: Makes it possible to visualize high-dimensional data in 2D or 3D.
- **Reduces Noise**: By discarding components with low variance, PCA removes irrelevant or noisy features.

Step 1: Standardize the Data

PCA requires data to be standardized (zero mean and unit variance). Let's compute the standardized values for the given dataset.

Data:

- Math Scores (x_1): $85, 78, 90, 45, 50, 40$
- English Scores (x_2): $70, 65, 88, 55, 50, 60$

Compute Means:

$$\mu_{x_1} = \frac{85 + 78 + 90 + 45 + 50 + 40}{6} = 64.67$$

$$\mu_{x_2} = \frac{70 + 65 + 88 + 55 + 50 + 60}{6} = 64.67$$

Compute Standard Deviations:

$$\sigma_{x_1} = \sqrt{\frac{\sum (x_{i1} - \mu_{x_1})^2}{n-1}}$$

$$\sigma_{x_2} = \sqrt{\frac{\sum (x_{i2} - \mu_{x_2})^2}{n-1}}$$

Using the formula, we compute:

$$\sigma_{x_1} = \sqrt{\frac{(85 - 64.67)^2 + (78 - 64.67)^2 + \cdots + (40 - 64.67)^2}{5}} = 21.12$$

$$\sigma_{x_2} = \sqrt{\frac{(70 - 64.67)^2 + (65 - 64.67)^2 + \cdots + (60 - 64.67)^2}{5}} = 12.72$$

Standardize the Data:

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

For each data point:

$$z_{11} = \frac{85 - 64.67}{21.12} = 0.96, \quad z_{12} = \frac{70 - 64.67}{12.72} = 0.42$$

Repeating for all data points, the standardized data is:

$$Z = \begin{bmatrix} 0.96 & 0.42 \\ 0.63 & 0.03 \\ 1.20 & 1.83 \\ -0.93 & -0.76 \\ -0.70 & -1.16 \\ -1.16 & -0.42 \end{bmatrix}$$

Step 2: Compute the Covariance Matrix

The covariance matrix is calculated using the standardized data:

$$\text{Cov}(x_1, x_2) = \frac{\sum (z_{i1} - \mu_{z_1})(z_{i2} - \mu_{z_2})}{n - 1}$$

Since the data is standardized ($\mu_{z_1} = \mu_{z_2} = 0$):

$$\text{Cov}(x_1, x_2) = \frac{\sum z_{i1} z_{i2}}{n-1}$$

Variances:

$$\text{Var}(x_1) = \frac{\sum z_{i1}^2}{n-1}, \quad \text{Var}(x_2) = \frac{\sum z_{i2}^2}{n-1}$$

Using the data:

$$\text{Var}(x_1) = 1, \quad \text{Var}(x_2) = 1, \quad \text{Cov}(x_1, x_2) = 0.88$$

The covariance matrix is:

$$\text{Covariance Matrix} = \begin{bmatrix} 1 & 0.88 \\ 0.88 & 1 \end{bmatrix}$$

Step 3: Compute Eigenvalues and Eigenvectors

Solve the characteristic equation:

$$\text{Covariance Matrix} \cdot v = \lambda v$$

The eigenvalues (λ) are roots of:

$$\det(\text{Covariance Matrix} - \lambda I) = 0$$

Expanding:

$$\det \begin{bmatrix} 1-\lambda & 0.88 \\ 0.88 & 1-\lambda \end{bmatrix} = (1-\lambda)^2 - (0.88)^2 = 0$$

$$(1-\lambda)^2 - 0.7744 = 0 \implies \lambda^2 - 2\lambda + 0.2256 = 0$$

Solving for λ:

$$\lambda_1 = 1.88, \quad \lambda_2 = 0.12$$

Eigenvectors:

Substitute $\lambda_1 = 1.88$ into:

$$\begin{bmatrix} 1-\lambda & 0.88 \\ 0.88 & 1-\lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

This gives:

$$v_1 = \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix}, \quad v_2 = \begin{bmatrix} -0.71 \\ 0.71 \end{bmatrix}$$

Step 4: Select Principal Components

The eigenvalue $\lambda_1 = 1.88$ explains:

$$\frac{\lambda_1}{\lambda_1 + \lambda_2} = \frac{1.88}{1.88 + 0.12} \approx 94\%$$

The eigenvalue $\lambda_2 = 0.12$ explains:

$$\frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{0.12}{1.88 + 0.12} \approx 6\%$$

Thus, we retain the first principal component (PC_1).

Step 5: Project Data onto Principal Components

The transformation matrix is:

$$V = \begin{bmatrix} 0.71 & -0.71 \\ 0.71 & 0.71 \end{bmatrix}$$

The projected data is:

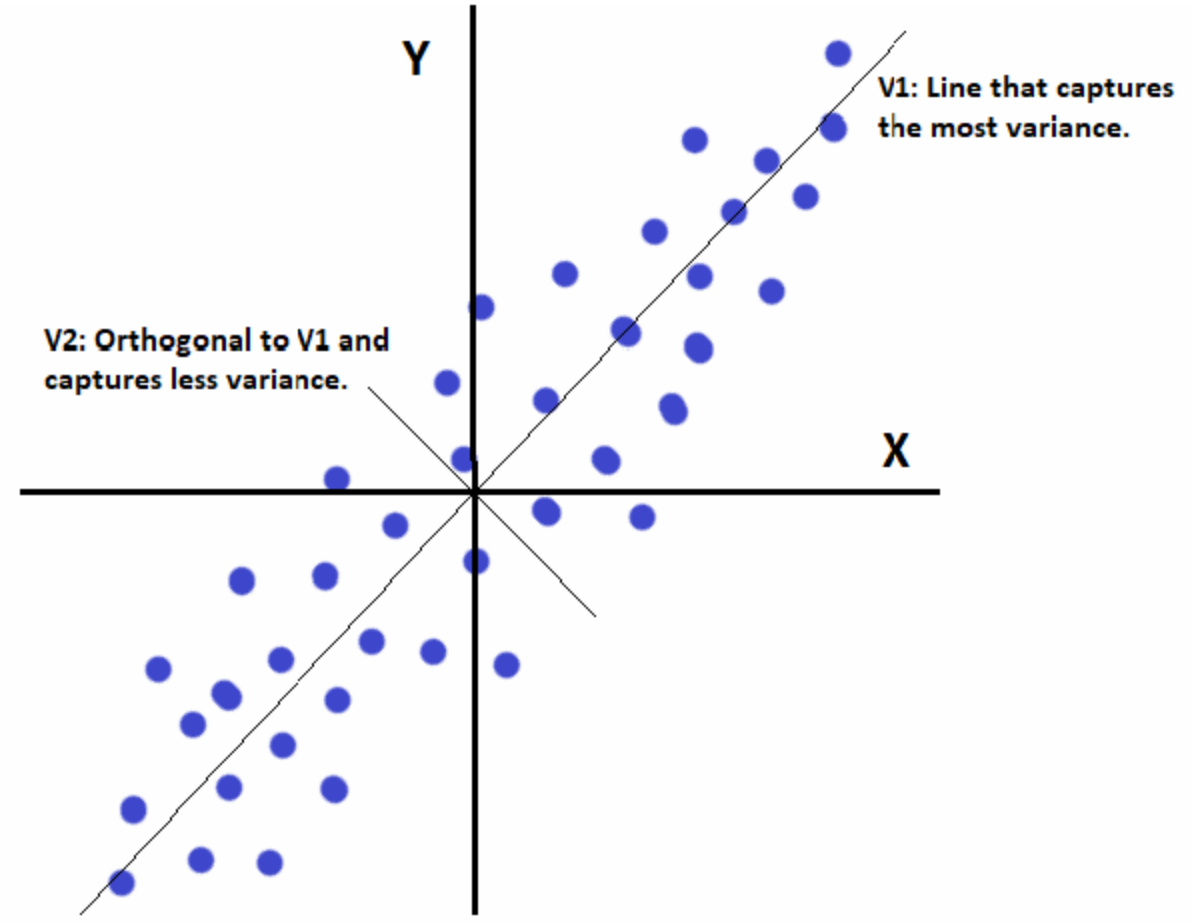
$$Z_{\text{projected}} = Z \cdot V$$

Performing the matrix multiplication yields the transformed data:

$$Z_{\text{projected}} = \begin{bmatrix} 1.16 & 0.02 \\ 0.47 & -0.42 \\ 2.55 & 0.45 \\ -1.19 & 0.12 \\ -1.32 & -0.10 \\ -1.67 & -0.07 \end{bmatrix}$$

Result

- Original dimensions: 2 (Math and English scores)
- Reduced dimensions: 1 (PC_1), retaining ~94% of the variance.



Summary of steps to perform PCA

1. Standardize the Data

The raw data X with n observations and p features is standardized to ensure all features contribute equally:

$$X_{\text{standardized}} = \frac{X - \mu}{\sigma}$$

where:

- μ: Mean of each feature.
- σ: Standard deviation of each feature.

---

2. Compute the Covariance Matrix

Calculate the covariance matrix Σ to measure the relationships between features:

$$\Sigma = \frac{1}{n-1} X_{\text{standardized}}^T X_{\text{standardized}}$$

Here, Σ is a $p \times p$ symmetric matrix where each element Σ_{ij} represents the covariance between features i and j.

---

3. Calculate Eigenvalues and Eigenvectors

Solve the eigenvalue equation:

$$\Sigma v_i = \lambda_i v_i$$

where:

- λ_i: Eigenvalues, representing the variance explained by the i-th principal component.
- v_i: Eigenvectors, representing the direction of the i-th principal component.

Eigenvalues and eigenvectors are computed for the covariance matrix. The eigenvectors form an orthogonal basis, and the eigenvalues indicate the importance of each basis vector.

---

4. Sort Eigenvalues and Select Top k Components

Rank the eigenvalues λ_i in descending order. The eigenvectors v_i corresponding to the largest eigenvalues capture the most variance.

Select the top k eigenvectors to form the transformation matrix V_k:

$$V_k = [v_1, v_2, \ldots, v_k]$$

Here, V_k is a $p \times k$ matrix.

5. **Project the Data**

Transform the original data into the new k-dimensional subspace:

$$Z = X_{\text{standardized}} \cdot V_k$$

where:

- Z: Transformed dataset with reduced dimensions ($n \times k$).
- V_k: Matrix of top k eigenvectors.

---

6. **Reconstruction (Optional)**

Approximate the original data using the reduced components:

$$X_{\text{reconstructed}} = Z \cdot V_k^T$$

This step helps validate the quality of dimensionality reduction by comparing the reconstructed data to the original.
