When is PCA Used?

PCA is a versatile technique applied in various scenarios:

1. **Better Perspective and Less Complexity:** When dealing with datasets containing a large number of features, it can be hard to grasp the underlying structure or build efficient models. If we have an intuition (or discover through analysis) that many features are redundant or that the core information can be captured in fewer dimensions, PCA helps simplify the data. Modeling is often much easier and more robust in 2D or 3D compared to hundreds of dimensions.
2. **Better Visualization:** As mentioned, visualizing high-dimensional data is impossible. PCA is frequently used to reduce the dimensionality down to 2 or 3 principal components, allowing us to create scatter plots or 3D plots that represent a "shadow" or projection of the original data, revealing clusters, trends, or outliers.
3. **Reduce Size / Speed Up Algorithms:** High-dimensional data increases storage requirements and slows down computation, especially for process-intensive algorithms (like many supervised learning methods, distance calculations, etc.). PCA reduces the number of features, leading to smaller data sizes, less redundancy, and faster training times.
4. **Different Perspective / Feature Engineering:** Even without the explicit need for reduction, PCA can provide a different perspective on the data. The principal components represent the directions of maximum variance and are linearly independent (uncorrelated) combinations of the original features. These new components can sometimes be used as engineered features themselves to describe the data differently or potentially improve the performance of subsequent models.

PCA Process Steps: The Workflow

Performing PCA involves a sequence of mathematical steps:

1. **Standardize the Scale of Initial Set of Features:** Ensure all features are on the same scale. This is crucial because PCA is sensitive to the variance of the features.
2. **Derive the Covariance Matrix:** Calculate the covariance matrix for the standardized features to understand how features vary with respect to each other.
3. **Derive Eigenvectors and Eigenvalues:** Compute the eigenvectors and eigenvalues of the covariance matrix. These reveal the principal components (directions of maximum variance) and the amount of variance captured by each.
4. **Create a Feature Vector (Select Principal Components):** Decide how many principal components to keep based on the eigenvalues (explained variance). Form a "feature vector" using the eigenvectors corresponding to the chosen components.
5. **Represent the Data along the Principal Components Axes:** Transform the original standardized data into the new coordinate system defined by the selected principal components (feature vector). This is the final lower-dimensional representation of the data.

Let's look at the initial steps in more detail.

Step 1: Standardize Dataset Features

**Why Standardize?** PCA identifies principal components based on the **variance** in the data. If features have vastly different scales (e.g., one feature ranges from 0-1, another from 0-1,000,000), the feature with the larger scale (and thus typically larger variance) will dominate the PCA calculation, and the principal components will be biased towards that feature. Standardization prevents this by giving all features equal importance initially.

* **Goal:** Bring uniformity to the scales of continuous numeric variables, typically transforming them to have a **mean of 0** and a **standard deviation of 1**.
* **Method (Z-score):** The most common method is Z-score standardization. For each value x in a feature column:
* x\_stand = (x - mean(x)) / standard\_deviation(x)

This is done by subtracting the mean of the feature column from each value and then dividing by the standard deviation of that column.

Standardization is a critical pre-processing step for PCA and many other algorithms sensitive to feature scales (like SVMs, K-Means, Gradient Descent-based algorithms).

Step 2: Derive Covariance Matrix

Once the data is standardized, the next step is to understand how the variables relate to each other in terms of their *co-variability*. This is captured by the **Covariance Matrix**.

* **Definition:** The covariance matrix is a square (p x p, where p is the number of features/dimensions) and symmetric matrix. Each entry (i, j) in the matrix represents the **covariance** between feature i and feature j.
* **Variance vs. Covariance:**
  + **Variance (var(x) or cov(x, x)):** Measures the spread or variability of a *single* feature around its mean. It's the square of the standard deviation and appears on the diagonal of the covariance matrix.
  + **Covariance (cov(x, y)):** Measures the degree to which *two* features (x and y) vary together (co-vary). It indicates the direction of the linear relationship between the variables.
* **Structure:**
  + For a dataset with 2 features (x, y):
  + [ var(x) cov(x, y) ]
  + [ cov(y, x) var(y) ]

(Note: cov(x, y) = cov(y, x))

* + For a dataset with 3 features (x, y, z):
  + [ var(x) cov(x, y) cov(x, z) ]
  + [ cov(y, x) var(y) cov(y, z) ]
  + [ cov(z, x) cov(z, y) var(z) ]
* **Calculation of Covariance:** The covariance between two variables x and y is calculated as:
* cov(x, y) = Σ [ (xᵢ - μₓ) \* (yᵢ - μ<0xE1><0xB5><0xB3>) ] / N (for i=1 to N)

Where xᵢ and yᵢ are individual data points, μₓ and μ<0xE1><0xB5><0xB3> are the means of variables x and y respectively, and N is the number of data points. (Sometimes N-1 is used in the denominator for an unbiased sample estimate, but the principle is the same).

* **Interpretation of Covariance Value:**
  + **Positive Covariance:** Indicates that as one variable tends to increase, the other variable also tends to increase (e.g., height and weight). Data points cluster along a positive slope.
  + **Negative Covariance:** Indicates that as one variable tends to increase, the other variable tends to decrease (e.g., price and demand). Data points cluster along a negative slope.
  + **Zero (or Near-Zero) Covariance:** Indicates little to no linear relationship between the two variables. Data points form a cloud with no clear directional trend.

The covariance matrix summarizes the variance within each feature and the linear relationships between all pairs of features. This matrix forms the basis for the next step.

Step 3: Derive Eigenvectors and Eigenvalues (Eigen-Decomposition)

The crucial step in PCA is to extract the directions of maximum variance and the magnitude of that variance from the covariance matrix. This is done through **Eigen-Decomposition**.

For any square matrix **A** (like our covariance matrix **C**), its **eigenvectors** (v) and **eigenvalues** (λ) are special vectors and scalars that satisfy the following equation:

**Av = λv**

In the context of the **Covariance Matrix (C)**:

* **Eigenvectors (v):** These represent the **directions** of the axes where the data has the most variance (spread). They form a new set of orthogonal (perpendicular) basis vectors for the data. These directions are the **Principal Components**. The first eigenvector corresponds to the direction of *maximum* variance, the second eigenvector corresponds to the direction of maximum *remaining* variance orthogonal to the first, and so on.
* **Eigenvalues (λ):** These are scalar coefficients associated with each eigenvector. Each eigenvalue represents the **amount of variance** carried or explained by its corresponding eigenvector (Principal Component). Larger eigenvalues correspond to principal components that capture more variance in the data.
* **Finding Principal Components:** By calculating the eigenvectors and eigenvalues of the covariance matrix (a process called Eigen-Decomposition) and **ranking the eigenvectors** in descending order based on their corresponding eigenvalues (highest to lowest), we obtain the Principal Components ordered by their significance (amount of variance explained).

This decomposition allows us to identify the new axes (Principal Components) that best represent the spread and structure of the data.