Dimensionality Reduction
Part 1

Mohammed Brahimi & Sami Belkacem

Dimensionality Reduction

Dimensionality reduction is a technique in data analysis that simplifies the data by reducing the number of variables or dimensions.

Dimensionality Reduction should ensure the following:

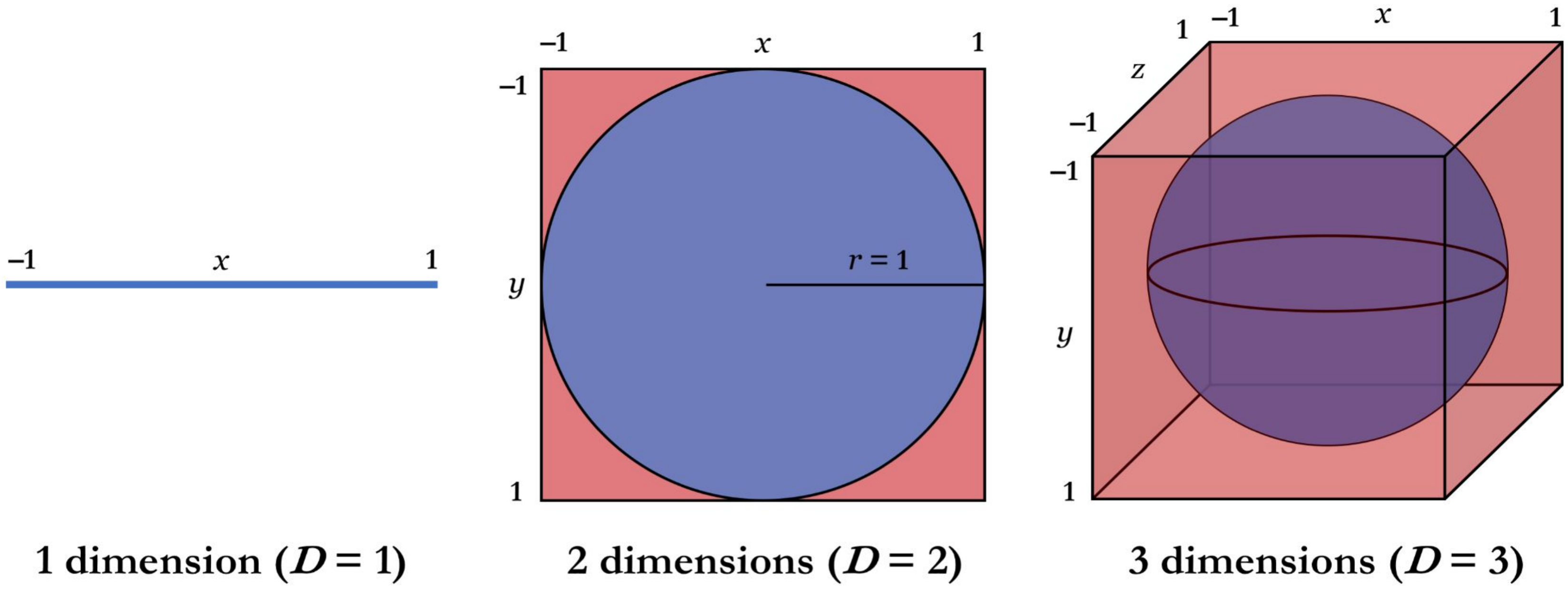
- **Preserving Relevance**
  - It retains the most crucial information from high-dimensional spaces.
- **Eliminating Redundancy**
  - Dimensionality reduction efficiently removes redundant and irrelevant attributes for the analysis.
- **Transformation into a lower-dimensional space**
  - it transforms data into a lower-dimensional space, making it more manageable for analysis and visualization.

Motivations for Dimensionality Reduction

- Improve model
  - Improved Model Performance
  - Reduce Overfitting
- Efficiency and Resource Management
  - Computational Efficiency
  - Memory and Storage Efficiency
- Interpretability and Understanding
  - Data Visualization
  - Enhanced Interpretability
- Data quality enhancement
  - Removal of Redundant Information
  - Noise Reduction

The Curse of Dimensionality

The Curse of Dimensionality

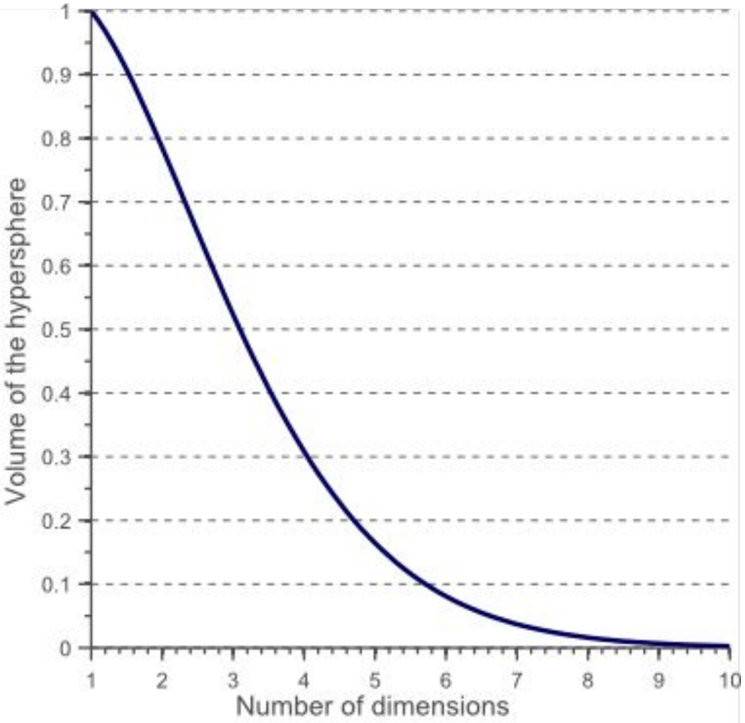


Volume of sphere VS Volume of cube

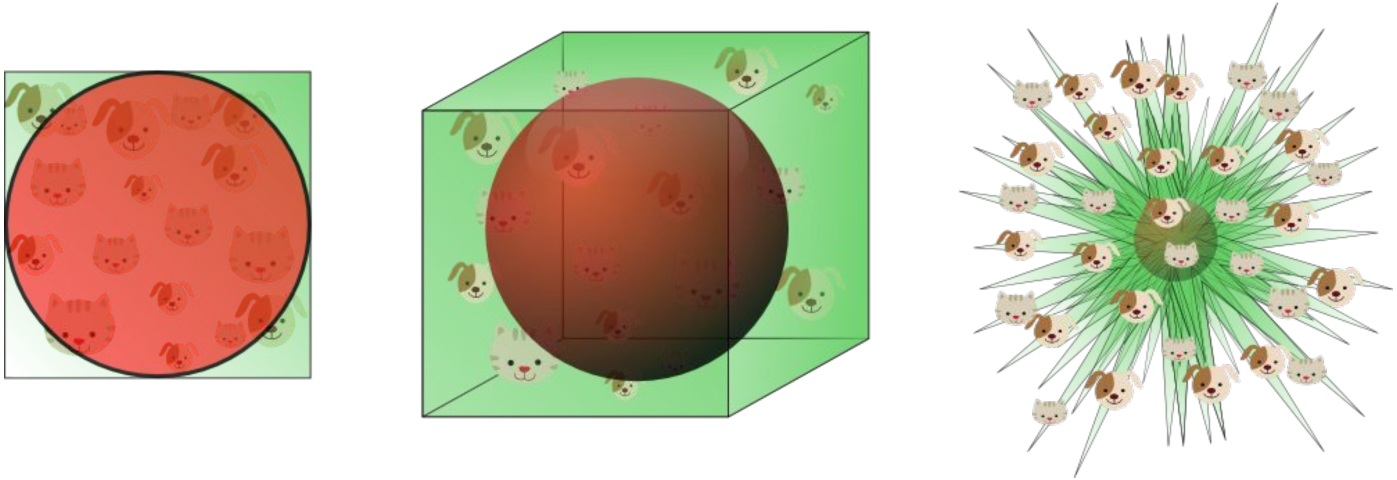
The Curse of Dimensionality

$$\frac{V_{\text{hypersphere}}}{V_{\text{hypercube}}} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} \to 0 \text{ as } d \to \infty.$$

As dimensionality (d) grows, the **hypersphere's volume** becomes negligible when compared to the **hypercube**.

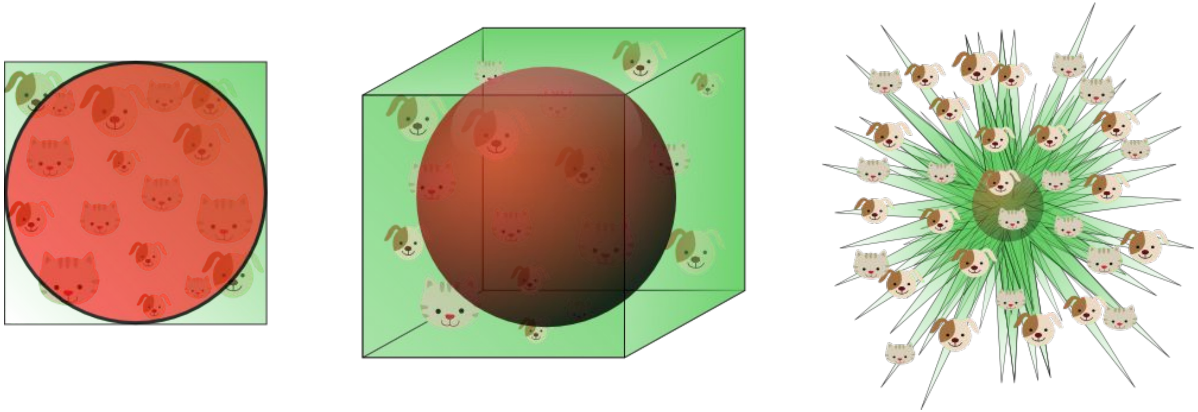


The Curse of Dimensionality



Sparsity: when points are uniformly generated within a high-dimensional hypercube, **most points are much farther from the center than expected.**

The Curse of Dimensionality



- Traditional Distance Metrics
  - In high dimensions, traditional distance metrics like Euclidean distance lose their effectiveness.
  - Points become dispersed, posing challenges for distance-based analysis.

The curse of dimensionality impacts diverse data analysis tasks (nearest-neighbor algorithms, classification, and clustering in high-dimensional spaces …).

Taxonomy of Dimensionality Reduction Methods

- **Feature extraction:** transforms the original attributes into new ones.
  - **Linear methods (PCA)**: Transform the original attributes into a new set of **linear** attributes.
  - **Non-linear methods (t-SNE):** Transform the original attributes into a new set of **non-linear** attributes.

- **Feature selection:** selects a subset of original attributes.
  - **Filter methods:** Select attributes based on their statistical properties.
  - **Wrapper methods:** Use the performance of a machine learning model to evaluate the goodness of selected attributes.

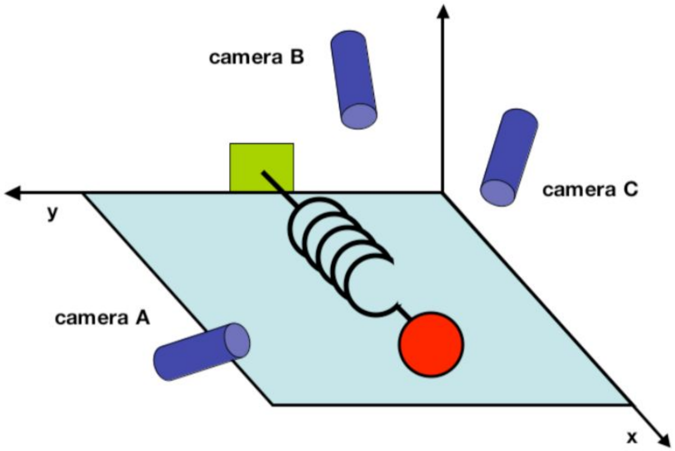
Outline

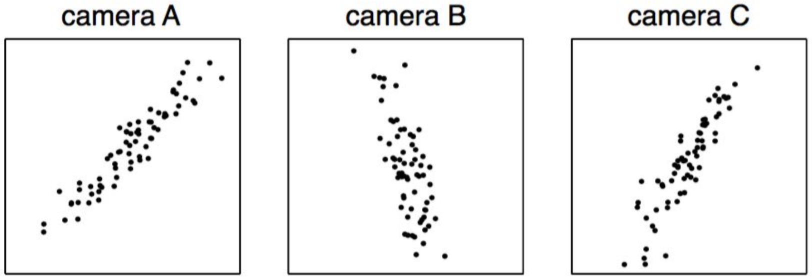
- **Principal Component Analysis (PCA)**
  - What is PCA?
  - Why PCA?
  - How to perform PCA?
  - Applications of PCA

Motivation example: oscillating spring

- Tracking motion of a ball on an oscillating spring at regular intervals.
- 3 cameras record (x,y) position from different angles
- Each sample is 6D vector:

  X = [xA, yA, xB, yB, xC, yC]



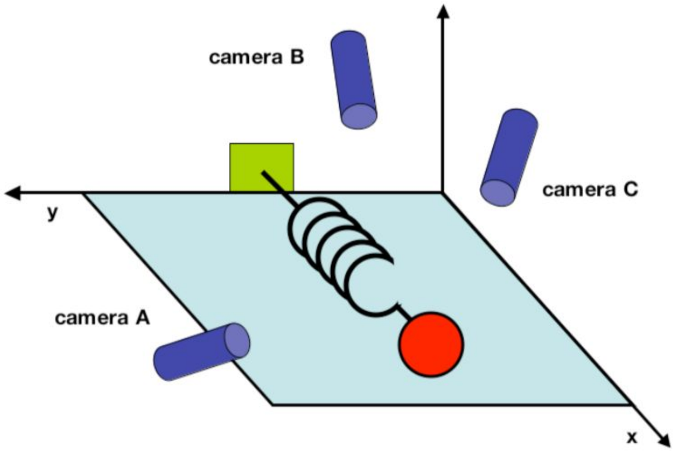


Do we need all six dimensions to study the motion of the spring?

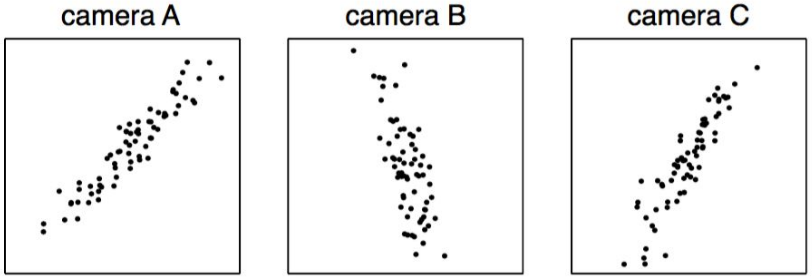
Motivation example: oscillating spring

- Tracking motion of a ball on an oscillating spring at regular intervals.
- 3 cameras record (x,y) position from different angles
- Each sample is 6D vector:

  X = [xA, yA, xB, yB, xC, yC]



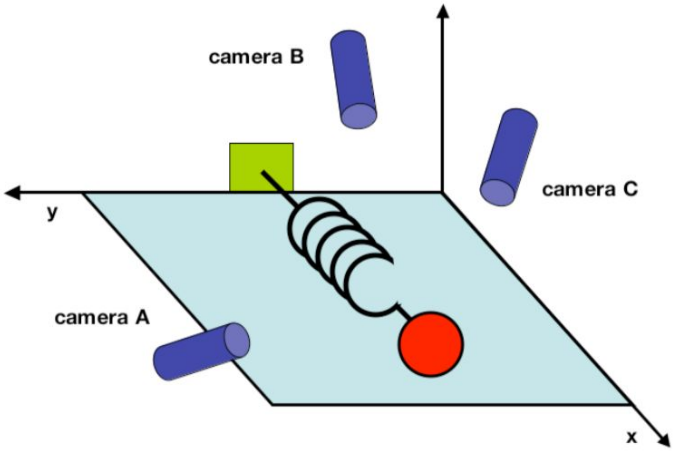
Physically, we know the underlying motion is 1D, along the spring.



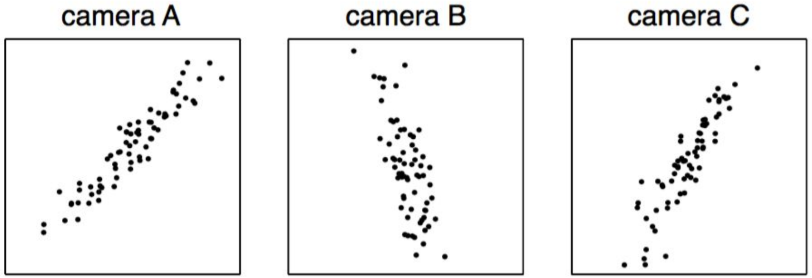
Motivation example: oscillating spring

- Tracking motion of a ball on an oscillating spring at regular intervals.
- 3 cameras record (x,y) position from different angles
- Each sample is 6D vector:

  X = [xA, yA, xB, yB, xC, yC]



How do we extract this underlying 1D dynamic hidden in 6D recordings ?



Correlations is common in Real Data (to be finished)

Correlation is a common characteristic of real-world datasets, often reflecting complex relationships.

Examples

- Weight and height correlation in individuals.
- Spatial pixel correlations in images.
- Study hours and test score relationships.

Data Redundancy

- Correlations can indicate redundancy in the data.
- Reducing redundancy can enhance the performance of data mining algorithms.

Recognizing and addressing correlations is essential for effective data analysis and machine learning applications.

What is Principal Components Analysis (PCA)?

PCA's goal

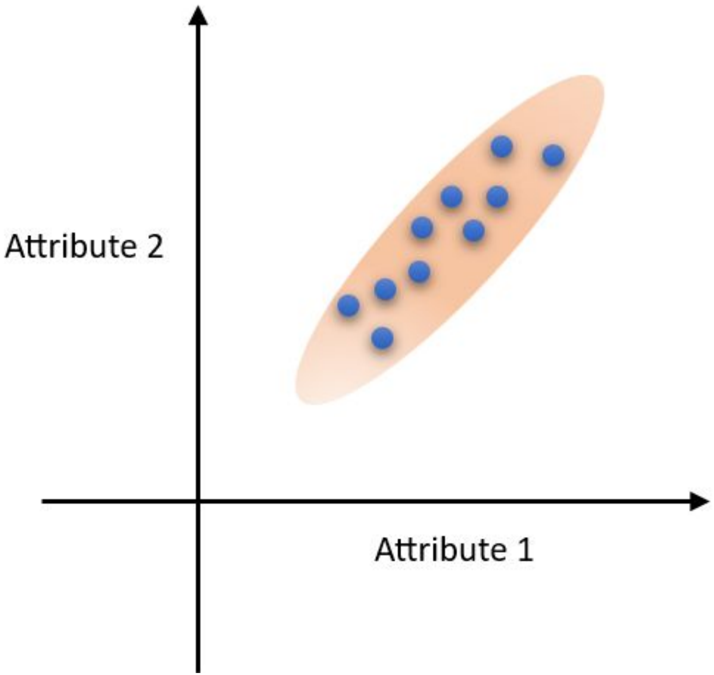
- Transform the original data into new variables that follow the **variation** in the data.

PCA's principle

- PCA rotates the axis to align with the direction of maximum data variability.

Principal Components

- After rotation, new axes, called principal components, are formed.
- The first principal component (PC1) captures the most variation.



What is Principal Components Analysis (PCA)?

PCA's goal

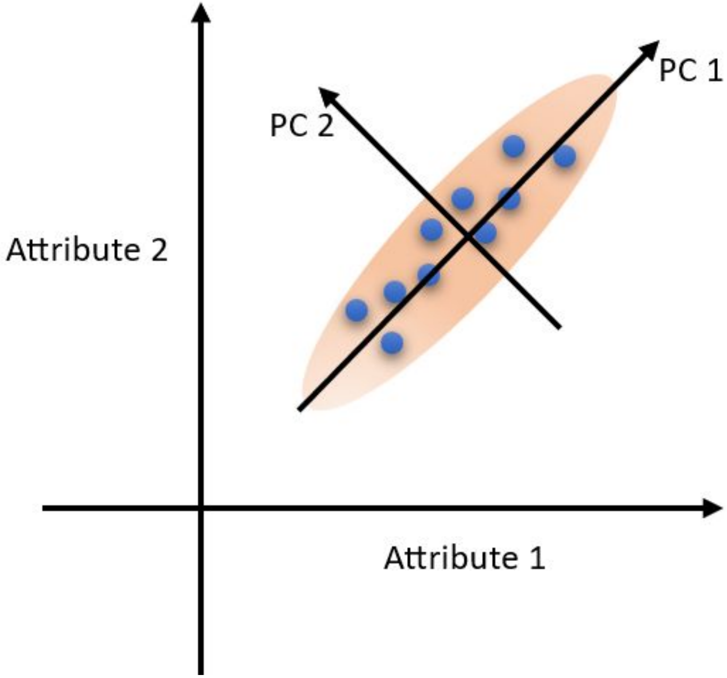
- Transform the original data into new variables that follow the **variation** in the data.

PCA's principle

- PCA rotates the axis to align with the direction of maximum data variability.

Principal Components

- After rotation, new axes, called principal components, are formed.
- The first principal component (PC1) captures the most variation.



What is Principal Components Analysis (PCA)?

PCA's goal

- Transform the original data into new variables that follow the **variation** in the data.

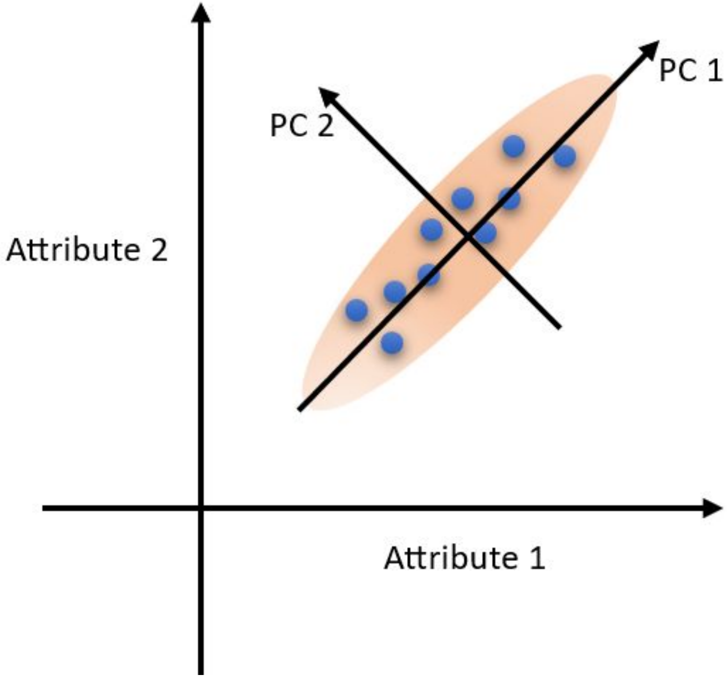
PCA'

- m

Using PC1, we reduce the data's dimensionality while retaining most of its variation.

Princ

- After rotation, new axes, called principal components, are formed.
- The first principal component (PC1) captures the most variation.



What is Principal Components Analysis (PCA)?

PCA's goal

- Transform the original data into new variables that follow the **variation** in the data.

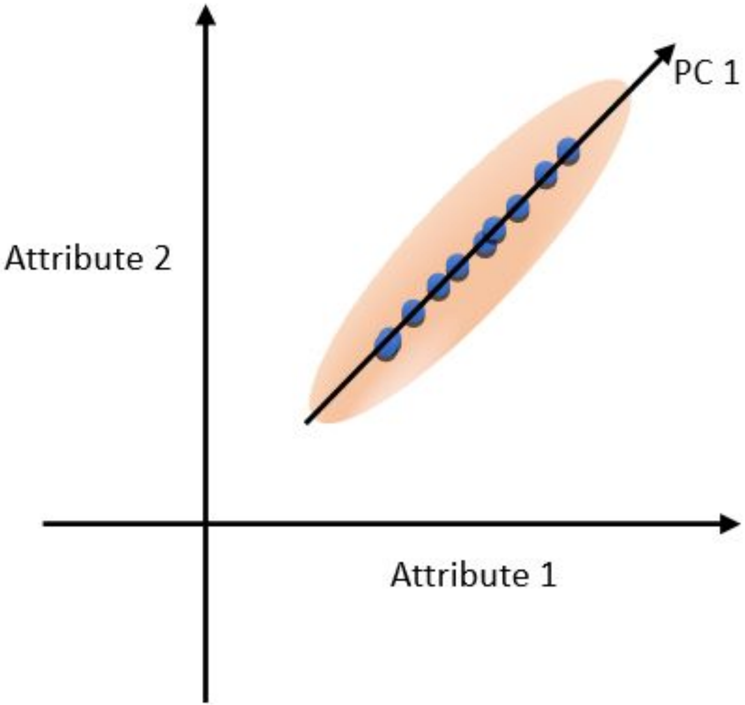
PCA'

- m

Princ

Using PC1, we reduce the data's dimensionality while retaining most of its variation.

- After rotation, new axes, called principal components, are formed.
- The first principal component (PC1) captures the most variation.



What is Principal Components Analysis (PCA)?

PCA's goal

- Transform the original data into new variables that follow the **variation** in the data.

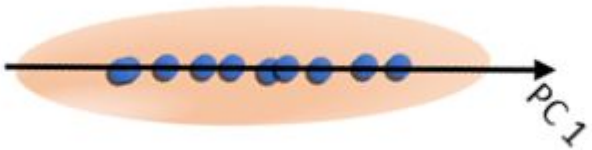
PCA'

- m

Prin

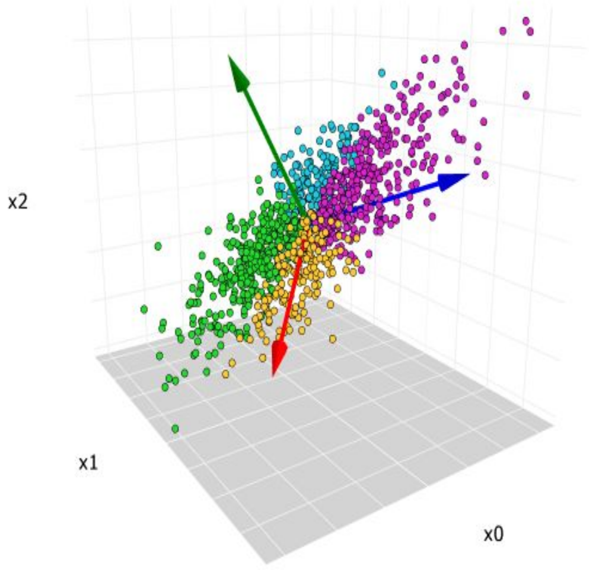
Using PC1, we reduce the data's dimensionality while retaining most of its variation.

- After rotation, new axes, called principal components, are formed.
- The first principal component (PC1) captures the most variation.



What is Principal Components Analysis (PCA)?

- PCA identifies the directions, called principal components, along which the data varies the most.
  - First principal component captures the most variance within the data.
  - Subsequent components are orthogonal (uncorrelated) to the preceding ones.
- Retaining only the most important components reduces the data's dimensionality.
- PCA is widely used in
  - Data compression, Visualization, Noise reduction, …

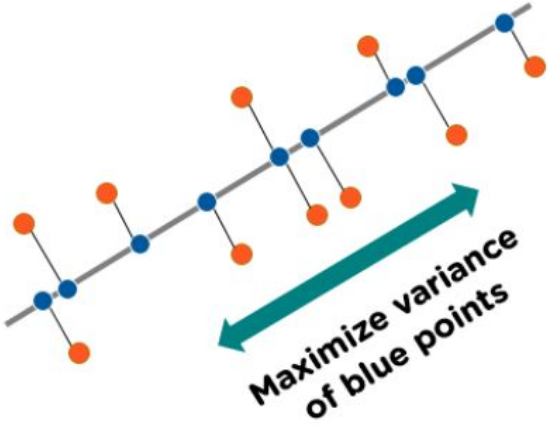


Objectives of PCA - A Mathematical Perspective

Maximizing Variance formulation

(Hotelling 1933)

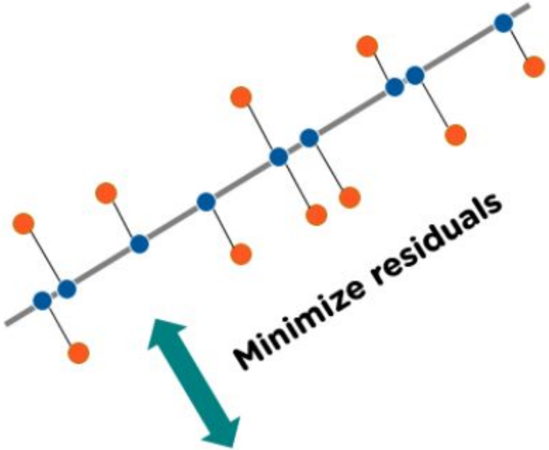
- Express the maximum **variation** within the first component.
- Subsequent principal components express the remaining variation.



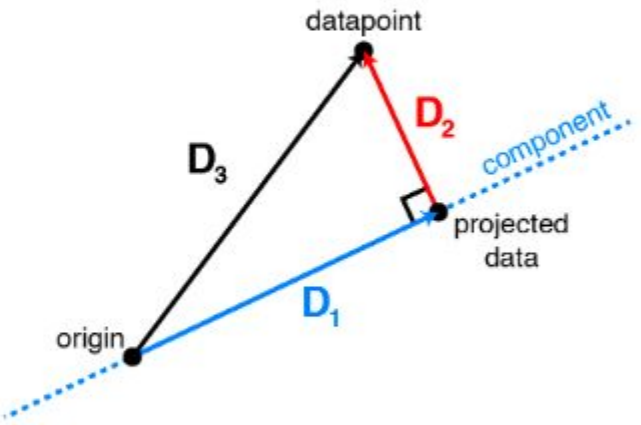
Minimizing error formulation

(Pearson 1901)

- Minimize the sum of projection **errors** on the first principal component.
- Successive components should also minimize projection errors.



Objectives of PCA - A Mathematical Perspective



$$D_3^2 = D_1^2 + D_2^2$$

initial variance = remaining variance + lost variance

$$\|\mathbf{a}_i\|^2 = \|w_i\,\mathbf{c}\|^2 + \|\mathbf{a}_i - w_i\,\mathbf{c}\|^2$$

this is constant | maximize this **or** minimize this

- **D1** is the remaining variance expressed in the component.
- **D2** is the projection error or the lost variance.
- **D3** is the original fixed variance independent to the component.

Maximizing the remaining variance (D1) is equivalent to minimizing the projection error (D2)

PCA Algorithm

- Step 1: Preprocessing
  - Subtract the mean from each attribute (column) to center the data.
  - Scale attributes if their scales differ.
- Step 2: Covariance Matrix
  - Calculate the covariance matrix, denoted as sigma.
- Step 3: Eigenvalues and Eigenvectors
  - Compute the eigenvalues and eigenvectors of the covariance matrix to find principal components.
- Step 4: Principal Components Selection
  - Choose **k** < **m** eigenvectors as principal components.
- Step 5: Dimensionality Reduction
  - Reduce data by projecting it onto the **k** principal components

The input of the algorithm

$$X = \begin{bmatrix} \vdots & \vdots & \cdots & \vdots \\ \mathbf{X}_1 & \mathbf{X}_2 & \cdots & \mathbf{X}_m \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix}_{n \times m}$$

n: is the number of objects (rows).

m: is the number of attributes (columns).

Covariance Matrix calculation

$$\mathrm{Cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{n-1}$$

Subtracting the mean from each attribute centers the data around zero, which makes the covariance formula easier to calculate:

$$\mathrm{Cov}(X,Y) = \frac{\sum_{i=1}^{n} X_i Y_i}{n-1}$$

Covariance Matrix calculation (Matrix notation)

$$\mathbf{\Sigma} = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \dots & \mathrm{Cov}(X_1, X_m) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \dots & \mathrm{Cov}(X_2, X_m) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_m, X_1) & \mathrm{Cov}(X_m, X_2) & \dots & \mathrm{Var}(X_m) \end{bmatrix}$$

This produce a symmetric covariance matrix :

$$\Sigma = \frac{1}{n-1} \begin{bmatrix} \dots & \mathbf{X}_1 & \dots \\ \dots & \mathbf{X}_2 & \dots \\ \vdots & \vdots & \vdots \\ \dots & \mathbf{X}_m & \dots \end{bmatrix} \begin{bmatrix} \vdots & \vdots & \dots & \vdots \\ \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_m \\ \vdots & \vdots & \dots & \vdots \end{bmatrix}$$

$$\mathbf{\Sigma} = \frac{1}{n-1} \mathbf{X}^T \mathbf{X}$$

Eigenvalues and Eigenvectors

Diagonalization in PCA

- A fundamental concept in PCA for simplifying the variance matrix.

$$\mathbf{\Sigma} = \mathbf{PDP}^T$$

- Diagonal matrix (**D**) consists of eigenvalues (λ).
- Eigenvectors are arranged in the matrix **P** enable data projection and dimensionality reduction.
- **Spectral theorem ensures eigenvectors' orthogonality.**
- **Singular value decomposition (SVD) can be used to do this decomposition.**

$$\mathbf{\Sigma} = \mathbf{USV}^T$$

Choosing k (number of principal components)

Select k<n eigenvectors to make them components

$$U = \begin{bmatrix} \underbrace{\begin{matrix} | & | & & | \\ u^{(1)} & u^{(2)} & \ldots & u^{(k)} \\ | & | & & | \end{matrix}}_{\textbf{Select k components}} & \begin{matrix} & | \\ \ldots & u^{(m)} \\ & | \end{matrix} \end{bmatrix} \in \mathbb{R}^{m \times m}$$

Choosing k (number of principal components)

- Average squared projection error

$$\frac{1}{n}\sum_{i=1}^{n}\left\|x^{(i)} - x_{\text{approx}}^{(i)}\right\|^2$$

- Total variation in the data

$$\frac{1}{n}\sum_{i=1}^{n}\left\|x^{(i)}\right\|^2$$

-
- Choose k to be smallest value so that

$$\frac{\frac{1}{n}\sum_{i=1}^{n}\left\|x^{(i)} - x_{\text{approx}}^{(i)}\right\|^2}{\frac{1}{n}\sum_{i=1}^{n}\left\|x^{(i)}\right\|^2} \leq 0.01$$

"99% of variance is retained"

Choosing k (number of principal components)

- Average squared projection error

$$\frac{1}{n}\sum_{i=1}^{n}\left\|x^{(i)} - x_{\text{approx}}^{(i)}\right\|^2$$

- Total variation in the data

$$\frac{1}{n}\sum_{i=1}^{n}\left\|x^{(i)}\right\|^2$$

-
- Choose k to be smallest value so that

$$1 - \frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \leqslant 0.01$$

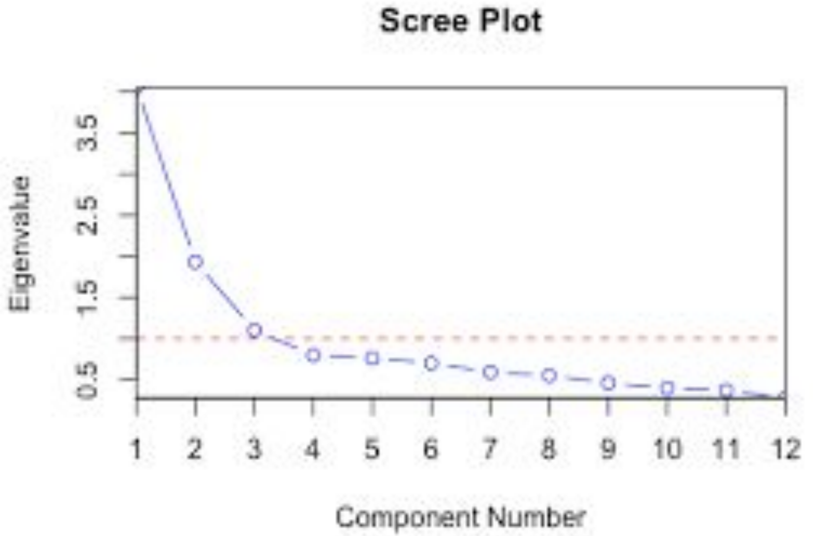
$$\frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \geq 0.99$$

"99% of variance is retained"

Choosing k (number of principal components)

- **Scree Plot**
  - Tool aids in deciding how many principal components to retain(k).

- **How to choose k**
  - Elbow point in the plot indicates a significant drop in eigenvalues, suggesting an optimal number of components.



Determining Coordinates in Principal Components

$$\mathbf{T} = \mathbf{XU}$$

- **T :** Coordinates matrix of data points in PCA components.
- **X :** Data matrix (variables as columns, data points as rows).
- **P :** Eigenvectors matrix of the covariance matrix.

For a dataset with m variables, to compute coordinates in the first k PCA components:

$$\mathbf{T_k} = \mathbf{XU}[:, 1 : \mathbf{k}]$$

Determining Coordinates in Principal Components

$$\mathbf{T} = \mathbf{X}\mathbf{U}$$

- **T :** Coordinates matrix of data points in PCA components.
- **X :** Data matrix (variables as columns, data points as rows).
- **P :** Eigenvectors matrix of the covariance matrix.

For a dataset with m variables, to compute coordinates in the first k PCA components:

$$\mathbf{T}_\mathrm{k} = \mathbf{X}\,\mathbf{U}_\mathrm{k}^\mathbf{T}$$

Reconstruction of data

$$\mathbf{X}_{\text{approx}} = \mathbf{T}_{\mathbf{k}} \mathbf{U}_{\mathbf{k}}^{\mathbf{T}}$$

- The matrix is approximation of the centred data.
- To reconstruct the original data, the mean should be added to each columns.

PCA applications: Denoising with PCA

Purpose

- Reduce noise in data or images.

How It Works

- Data is represented as a matrix.
- PCA identifies principal components.
- High-variance components retain signal;
- low-variance components capture noise.

Use Cases:

- Image denoising.
- Enhancing data quality in various fields.

PCA applications: Face Recognition with PCA

Purpose

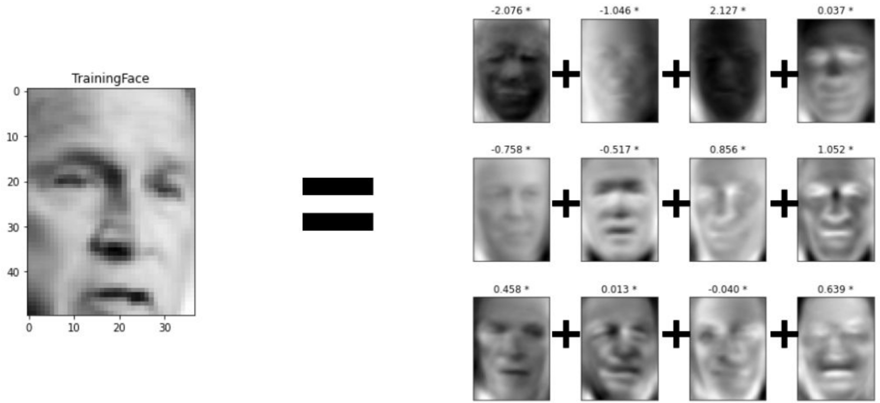
- Identify and authenticate individuals from facial images.

How It Works

- Eigenfaces: Eigenvalue and eigenvector decomposition of face images.
- Reduced-dimension representation.
- Compare face features for recognition.

Use Cases

- Security systems.
- Biometric authentication.
- Video surveillance.



PCA applications: Data Visualization with PCA

Purpose

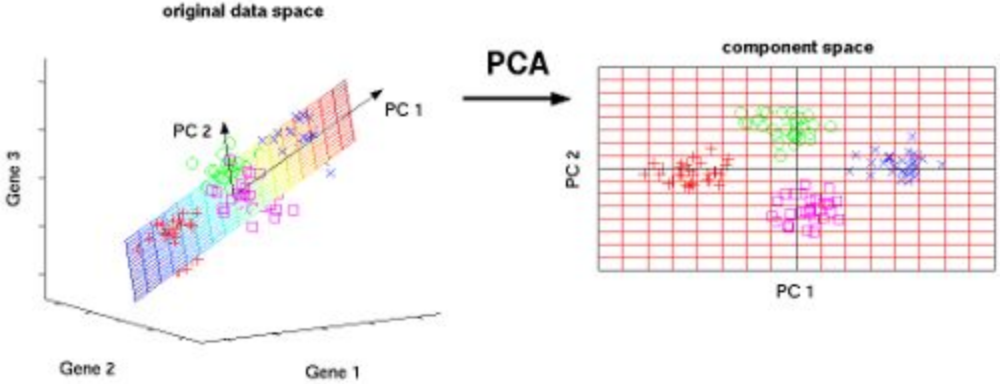
Transform high-dimensional data into a lower-dimensional space for visualization.

How It Works

- PCA reduces data dimensions while preserving data variance.
- Data points are projected onto a lower-dimensional (2D,3D) subspace.

Use Cases

- Exploratory data analysis.
- Visualizing multidimensional data in two or three dimensions.
- Cluster analysis.



PCA applications: Other

Versatility

- PCA is used in various fields and applications.

Examples

- **Anomaly Detection:** Identifying unusual data patterns.
- **Data Compression:** Reducing data dimensionality.
- **Recommendation Systems:** Extracting user preferences.
- **Genomic Data Analysis:** Identifying gene expression patterns.