# Data Mining For Decision Making

## Principal Components Analysis (PCA)

Prof. Varun Dutt

School of Computing and Electrical Engineering
Indian Institute of Technology Mandi, India

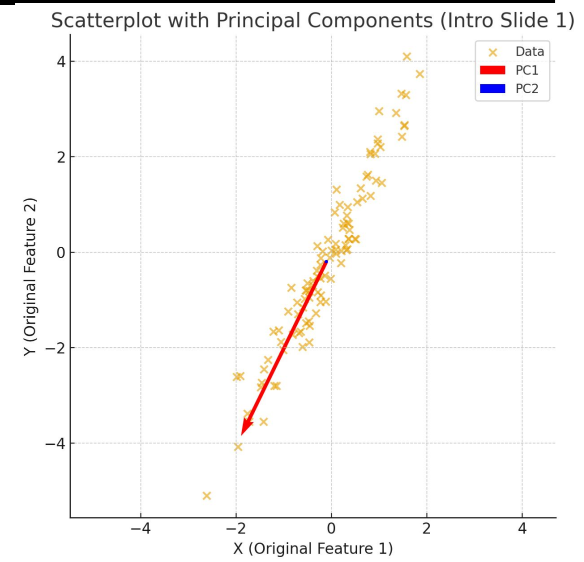
Indian Institute of Technology Mandi

Scaling the Heights

# Introduction to PCA

**Key Ideas:**

- PCA = a statistical technique for:
  - **Dimensionality reduction** (fewer variables, less redundancy).
  - **Feature extraction** (new meaningful variables = principal components).
- It re-expresses data along new coordinate axes that:
  - Are **uncorrelated** (orthogonal).
  - Capture **maximum variance** in descending order.
- Helps simplify data without losing much information.




**Why PCA?**

- Many datasets have correlated variables (e.g., height & weight, pixel intensities in images).
- PCA transforms correlated variables → uncorrelated "principal components".
- First few PCs often capture most of the information.

- Blue points = correlated dataset.
- Red arrow = **PC1 (direction of max variance)**.
- Blue arrow = **PC2 (orthogonal direction, less variance)**.
- This gives a perfect introduction to PCA as "rotating axes to capture maximum spread".

**Applications:**

- Face recognition (eigenfaces).
- Image compression.
- Visualization of high-dimensional data.
- Noise reduction.

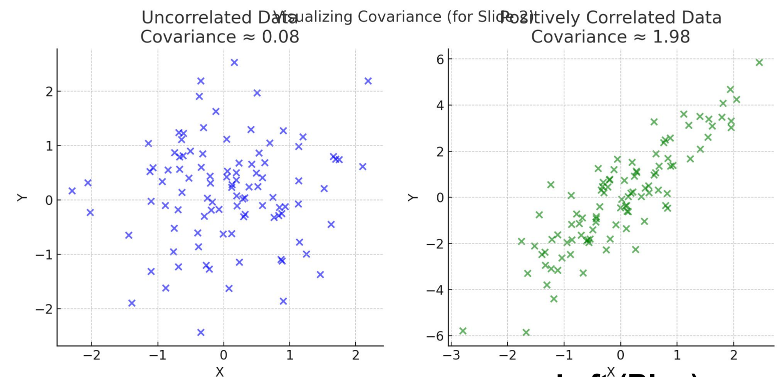
# Variance, Covariance, and Covariance Matrix

**Variance (σ²)**

- Measures how spread out a variable is around its mean.
- Formula:

$$Var(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

- Example: If test scores vary widely, variance is high.




**Left (Blue):** Uncorrelated data → covariance ≈ 0.
**Right (Green):** Positively correlated data → covariance > 0.

**Covariance Matrix**

- A square matrix that summarizes variance and covariance of all variables.

For 2D data:

$$\Sigma = \begin{bmatrix} Var(X) & Cov(X,Y) \\ Cov(X,Y) & Var(Y) \end{bmatrix}$$

- **Diagonal entries** = variances.
- **Off-diagonal entries** = covariances.

**Covariance (cov(X,Y))**

- Measures how two variables change together.
- Formula:

$$Cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

- Interpretation:
  - **Positive** → X and Y increase together.
  - **Negative** → one increases, other decreases.
  - **Zero** → no linear relationship.

## Why Important for PCA?

- PCA rotates the data so that axes (principal components) align with directions of **maximum variance.**
- Covariance matrix is the starting point for finding these directions.

# PCA Workflow (Step-by-Step)

### Step 1: Collect the Data

- Organize dataset into an $n \times d$ matrix (n = samples, d = features).
- Example: 10 points in 2D → 10 × 2 matrix.

### Step 2: Mean Center the Data

- Subtract the mean of each feature → new data with mean = 0.
- Ensures PCA is not biased by absolute position.

### Step 3: Compute Covariance Matrix

- Build covariance matrix Σ to capture relationships between features.
- Σ dimension = $d \times d$.

### Step 4: Find Eigenvalues & Eigenvectors

- Solve equation:

$$\Sigma v = \lambda v$$

- Eigenvectors (v) = directions of new axes (principal components).
- Eigenvalues (λ) = variance captured along each axis.

### Step 5: Sort by Variance

- Rank eigenvectors by descending eigenvalues.
- First component = PC1 (maximum variance).
- Second component = PC2, and so on.

### Step 6: Choose k Components

- Keep top $k$ eigenvectors → reduce dimensionality.
- Balance between dimensionality reduction and information loss.

### Step 7: Transform the Data

- New data = $FeatureVector^T \times MeanAdjustedData^T$.
- Original data → expressed in terms of principal components.

# Step 1: Example Dataset & Mean Centering

**Original Dataset (10 points in 2D)**

| x | y |
|---|---|
| 2.5 | 2.4 |
| 0.5 | 0.7 |
| 2.2 | 2.9 |
| 1.9 | 2.2 |
| 3.1 | 3.0 |
| 2.3 | 2.7 |
| 2.0 | 1.6 |
| 1.0 | 1.1 |
| 1.5 | 1.6 |
| 1.1 | 0.9 |

**Step 1: Compute the Mean**

- Mean of x:

$$\bar{x} = 1.81$$

- Mean of y:

$$\bar{y} = 1.91$$

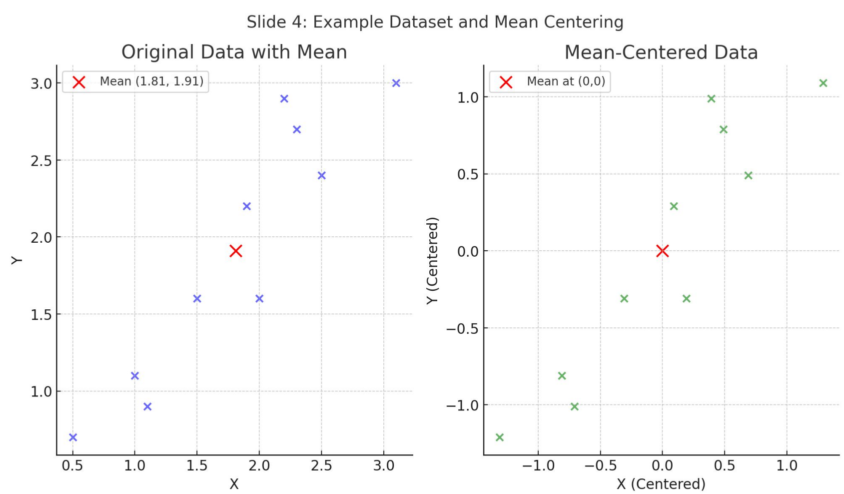
## Step 2: Mean-Center the Data

Subtract the mean from each value.

- Example: (2.5, 2.4) → (0.69, 0.49).
- After transformation, dataset has **mean = (0,0)**.

**Why Mean Centering?**

- Moves the dataset to the origin.
- Ensures PCA finds directions of maximum variance independent of absolute location.




**Left:** Original dataset with mean point marked (red X at (1.81, 1.91)).

**Right:** Mean-centered dataset (shifted so mean is at the origin).

# Step 2: Covariance Matrix

### Definition

For 2D data:

$$\Sigma = \begin{bmatrix} Var(X) & Cov(X,Y) \\ Cov(X,Y) & Var(Y) \end{bmatrix}$$

Where:

- $Var(X)$ = variance of feature X
- $Var(Y)$ = variance of feature Y
- $Cov(X,Y)$ = how X and Y vary together

### Computation (from handout dataset)

Using mean-centered data:

- $Var(X) = 0.6166$
- $Var(Y) = 0.7166$
- $Cov(X,Y) = 0.6154$

Thus:

$$\Sigma = \begin{bmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{bmatrix}$$

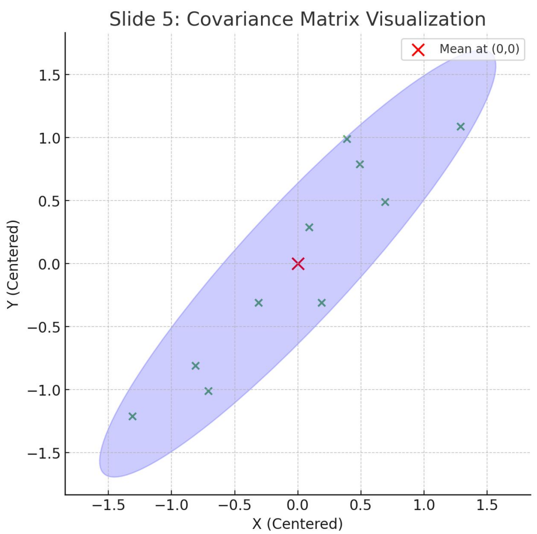
### Interpretation

- Positive covariance → X and Y increase together.
- Diagonal entries (variances) show spread of each feature.
- Off-diagonal entry shows correlation strength.

Green points = mean-centered dataset.

Red X = mean at (0,0).

Blue ellipse = covariance ellipse (spread and correlation).



# Step 3: Eigenvalues & Eigenvectors

## Eigenvalue Equation

We solve:

$$\Sigma v = \lambda v$$

For covariance matrix:

$$\Sigma = \begin{bmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{bmatrix}$$

### Step 1: Characteristic Polynomial

$$\det(\Sigma - \lambda I) = 0$$

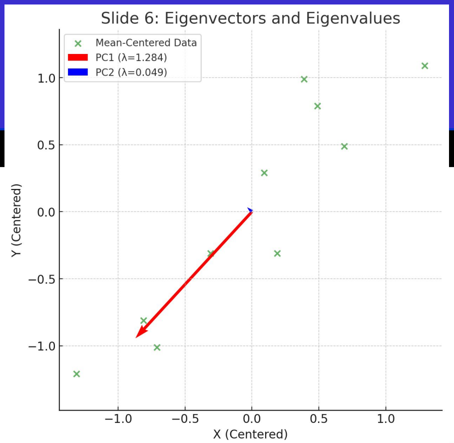
$$\det \begin{bmatrix} 0.6166 - \lambda & 0.6154 \\ 0.6154 & 0.7166 - \lambda \end{bmatrix} = 0$$

$$(0.6166 - \lambda)(0.7166 - \lambda) - (0.6154)^2 = 0$$

### Step 2: Expand

$$\lambda^2 - (0.6166 + 0.7166)\lambda + (0.6166)(0.7166) - (0.6154)^2 = 0$$

$$\lambda^2 - 1.3332\lambda + 0.0630 = 0$$




Green points = mean-centered dataset.

**Red arrow (PC1)** = direction of maximum variance ($\lambda_1 = 1.284$).

**Blue arrow (PC2)** = orthogonal minor component ($\lambda_2 = 0.049$).

### Step 3: Solve Quadratic

$$\lambda = \frac{1.3332 \pm \sqrt{1.3332^2 - 4(0.0630)}}{2}$$

$$\lambda_1 = 1.284, \quad \lambda_2 = 0.049$$

### Step 4: Find Eigenvectors

For $\lambda_1 = 1.284$:

$$\begin{bmatrix} -0.6674 & 0.6154 \\ 0.6154 & -0.5674 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

Solution: $y = 1.085x$ → eigenvector $v_1 = (-0.678, -0.735)$.

For $\lambda_2 = 0.049$:

Solution: $y = -0.923x$ → eigenvector $v_2 = (-0.735, 0.678)$.

## Interpretation

- **PC1 ($\lambda_1$=1.284, $v_1$)**: direction of maximum variance (~96%).
- **PC2 ($\lambda_2$=0.049, $v_2$)**: orthogonal, minor variance (~4%).

# Step 4: Projection onto Principal Components

### Why Project?

- We want to express the original data in terms of the new axes (principal components).
- This gives a rotated coordinate system where:
  - PC1 = maximum variance direction.
  - PC2 = orthogonal direction with minimal variance.

### Transformation Equation

If $X$ = mean-centered data matrix,

and $W$ = matrix of eigenvectors (columns),

then transformed data:

$$Y = XW$$

- Each row of $Y$ = new coordinates of a data point in PC space.
- If we keep only PC1 → 1D representation (dimensionality reduction)

**Example (Handout Data)**

Original (mean-centered point):

$(0.69, 0.49)$

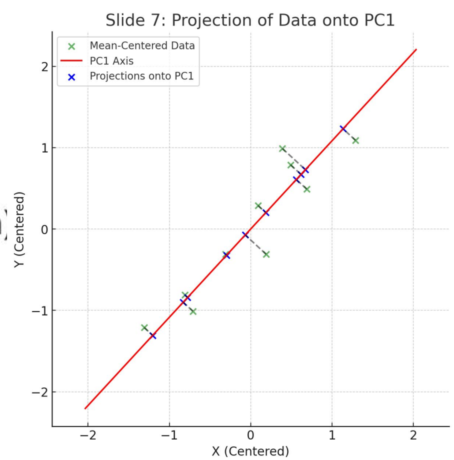
Dot product with PC1 $(-0.678, -0.735)$:

$$0.69(-0.678) + 0.49(-0.735) = -0.827$$

So in PC1 space, the point projects to ≈ **−0.827**.

### Interpretation

- Data "shadow" onto PC1 captures **most variance (~96%)**.
- Using only PC1: reduces 2D → 1D while retaining main pattern.
- Using both PCs: exact reconstruction.




**Green points:** Original mean-centered data.

**Red line:** PC1 axis (direction of maximum variance).

**Blue points:** Projections of data onto PC1.

**Dashed lines:** Show how each original point is projected down to PC1.

# Step 5: Variance Explained by PCs

### Eigenvalues Recap

- Each eigenvalue $\lambda$ = variance captured along its eigenvector (PC).
- Larger $\lambda$ → more important component.

From our dataset:

- $\lambda_1 = 1.284$ (PC1)
- $\lambda_2 = 0.049$ (PC2)

### Explained Variance Ratio

$$\text{Explained Variance Ratio} = \frac{\lambda_i}{\sum \lambda}$$

- PC1:

$$\frac{1.284}{1.284 + 0.049} = 0.963\ (96.3\%)$$

- PC2:

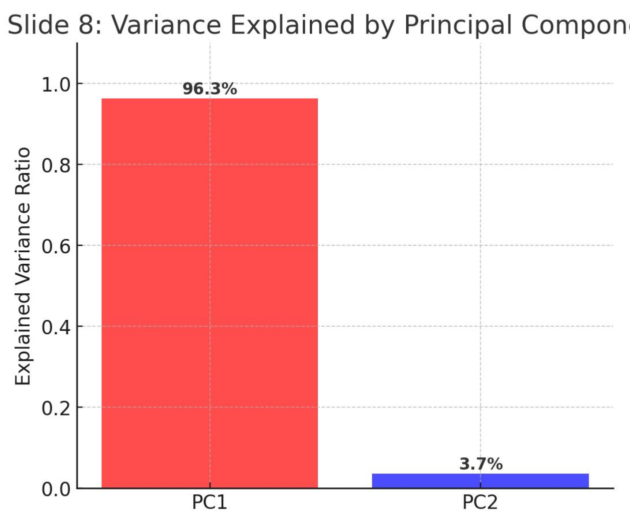
$$\frac{0.049}{1.284 + 0.049} = 0.037\ (3.7\%)$$

## Interpretation

- PC1 captures almost all of the variance (≈96%).
- PC2 adds very little information (≈4%).
- So projecting onto PC1 alone still preserves most structure in data.

### Applications

- **Dimensionality Reduction:**
  - Keep only PCs with large eigenvalues.
  - Discard PCs with tiny eigenvalues (mostly noise).
- **Data Compression:** Store fewer dimensions with minimal info loss.




**PC1 (red):** explains **96.3%** of the variance.

**PC2 (blue):** explains only **3.7%**.

# Step 6: Transforming Data to PC Space

## Transformation Equation

$$Y = XW$$

- $X$: mean-centered dataset
- $W$: eigenvector matrix (columns = eigenvectors)
- $Y$: transformed dataset in terms of principal components

## Example (Handout Data)

For point (0.69, 0.49):

- Projection onto PC1: ≈ −0.827
- Projection onto PC2: ≈ −0.175

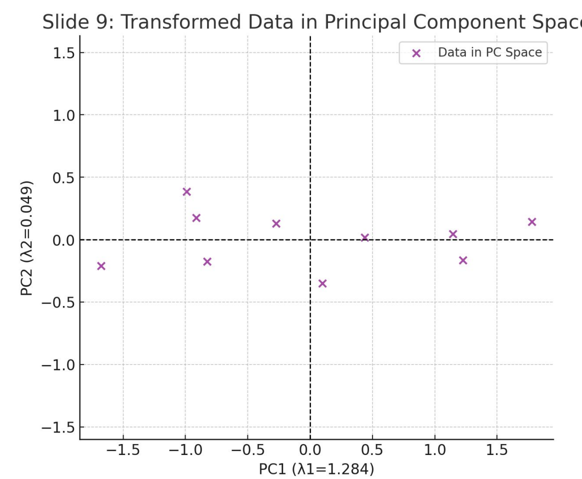
So in PC space: (−0.827, −0.175).

## PC Space

- New coordinates = linear combinations of original features.
- Axes are now **uncorrelated**.
- First axis (PC1) shows most variance.

**Interpretation**

- Scatterplot of data in PC1–PC2 space looks "uncorrelated."
- PC1 axis is stretched → dominant.
- PC2 axis is compressed → small variance.




- Scatterplot of the dataset in **PC space (PC1 vs PC2)**.
- Most variance is clearly along the **PC1 axis**, while **PC2 shows very little spread**.
- Confirms that dimensionality reduction to PC1 alone is effective.

# Step 7: Reconstruction from PCs

## Full Reconstruction (PC1 + PC2)

- If we use **all PCs**, the transformation is invertible:

$$X \approx YW^T + \text{mean}$$

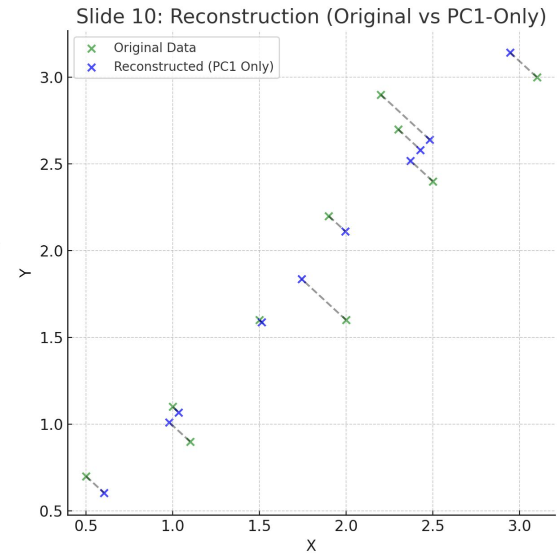
- This gives back the **original dataset exactly**.

## Reduced Reconstruction (PC1 only)

- Keep only PC1 → project data to 1D, then map back:

$$X_{approx} \approx Y_{PC1} \cdot W_{PC1}^T + \text{mean}$$

- Data lies along the PC1 line.




**Green points:** Original dataset.

**Blue points:** Reconstruction using **PC1 only**.

**Dashed lines:** Show the small error (distance lost along PC2).

## Example

Point (0.69, 0.49):

- Projection on PC1: −0.827.
- Reconstructed using PC1 only: ≈ (0.57, 0.62).
- Close to original, but slightly shifted toward PC1 line.

### Interpretation

- Using **all PCs**: no information loss.
- Using **PC1 only**: big compression, small loss.
- For this dataset: PC1 already captures **96% of the variance** → little information lost.

# Applications of PCA

## 1. Data Compression

- Reduce dimensionality while keeping most information.
- Example:
  - Images → keep only top PCs (“eigenimages”).
  - Store fewer numbers, yet image is still recognizable.

## 2. Noise Reduction

- Small eigenvalues often capture random noise.
- By discarding them, PCA denoises data.
- Example: ECG or EEG signals — remove noisy components.

## 3. Pattern Recognition

- PCA reveals underlying structure.
- **Face Recognition (Eigenfaces):**
  - Each face = combination of principal components (eigenfaces).
  - Compare faces in reduced PC space → faster, robust recognition.

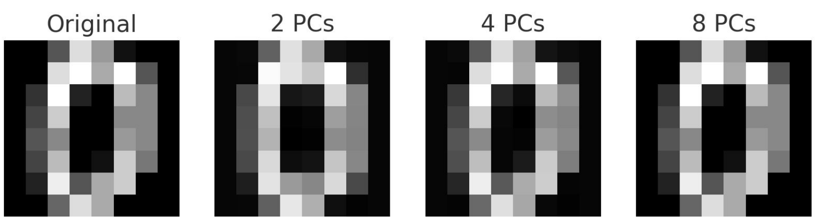
## 4. Data Visualization

- High-dimensional data → reduce to 2D or 3D.
- Example: visualizing gene expression profiles or word embeddings.

## Key Takeaway

- PCA is not just math → it’s a powerful tool for:
  - **Simplification**
  - **Compression**
  - **Visualization**
  - **Recognition**

Slide 11: PCA for Image Compression




**Left:** Original 8×8 digit image.

**Next:** Reconstructions with only **2 PCs, 4 PCs, and 8 PCs**.

Shows how fewer principal components still capture the main structure, but with some loss in detail.

# PCA: Summary & Key Points

## Key Insights

- **Principal Components (PCs):**
  - New orthogonal axes capturing max variance.
- **Eigenvalues:** tell how much variance each PC explains.
- **Explained Variance Ratio:** helps decide how many PCs to keep.
- **Dimensionality Reduction:** keep only top PCs → smaller, simpler dataset.

## Takeaway Message

- PCA = **rotate and compress** data.
- Captures essential structure while reducing complexity.
- Widely used in **ML, data science, and pattern recognition**.