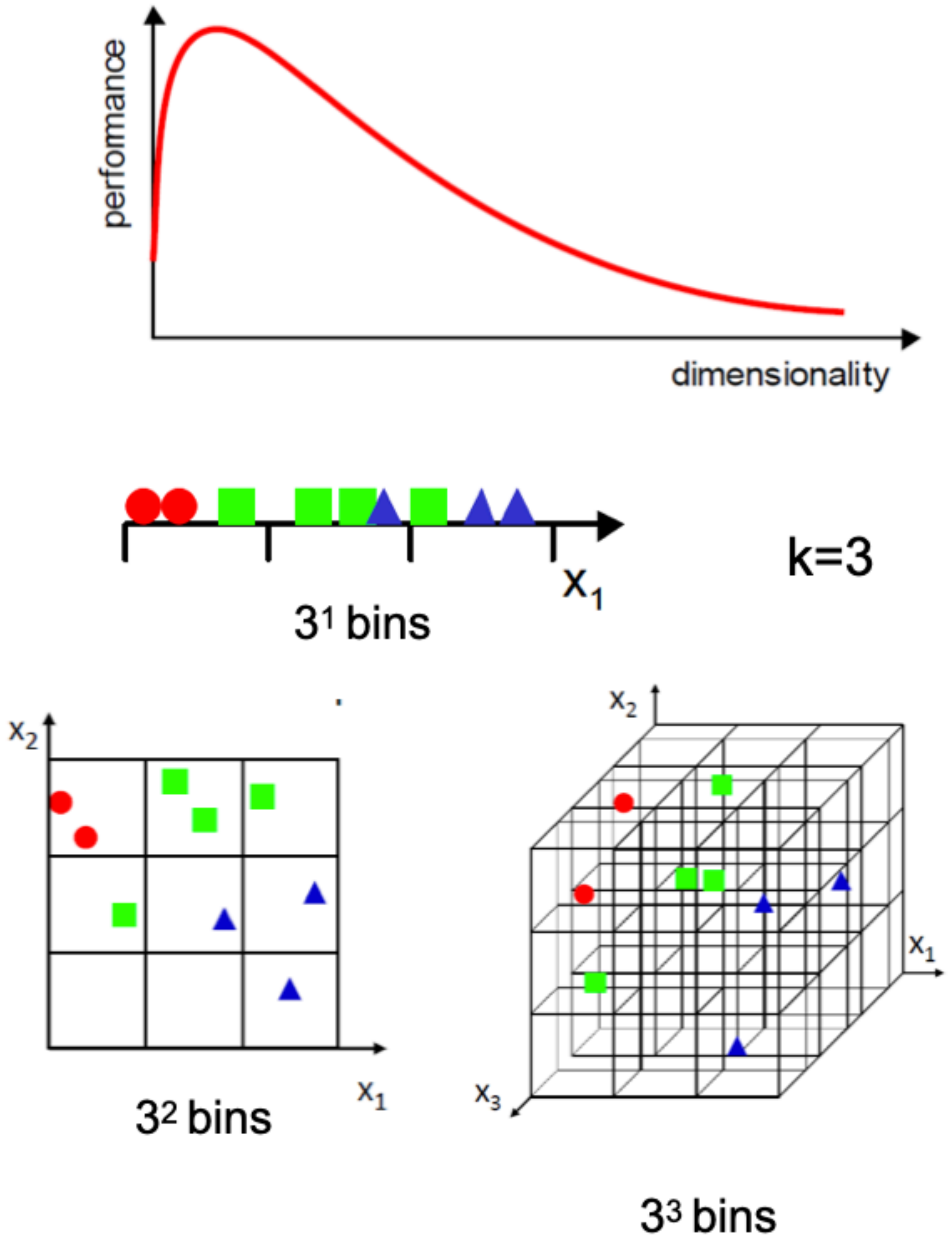
Principal Component Analysis

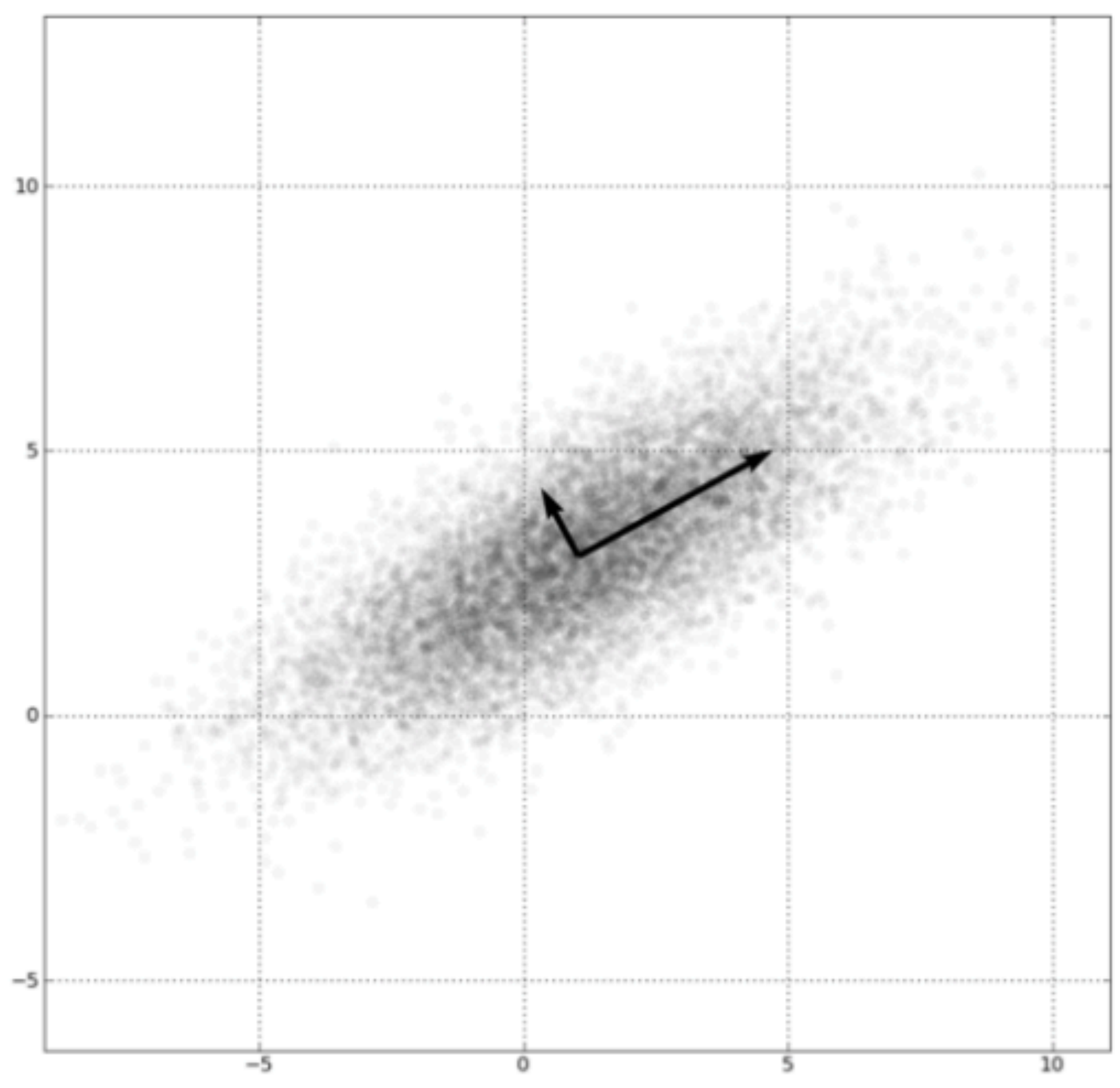
Curse of Dimensionality

- Increasing the number of features will not always improve classification accuracy.
- In practice, the inclusion of more features might actually lead to worse performance.
- The number of training examples required increases exponentially with dimensionality **d** (i.e., k^d).



Dimensionality Reduction

- Reduce high dimensional to lower dimensional space
- Preserve as much of variation as possible
- Plot lower dimensional space



The Idea: A toy example

- Consider the following 3D points
- If each component is stored in a byte, we need 18 = 3 x 6 bytes

1	2	4	3	5	6
2	4	8	6	10	12
3	6	12	9	15	18

The Idea: A toy example

- Looking closer, we can see that all the points are related geometrically: they are all the same point, scaled by a factor

$$\begin{bmatrix}1\\2\\3\end{bmatrix} = 1 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}2\\4\\6\end{bmatrix} = 2 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}4\\8\\12\end{bmatrix} = 4 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\begin{bmatrix}3\\6\\9\end{bmatrix} = 3 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}5\\10\\15\end{bmatrix} = 5 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}6\\12\\18\end{bmatrix} = 6 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

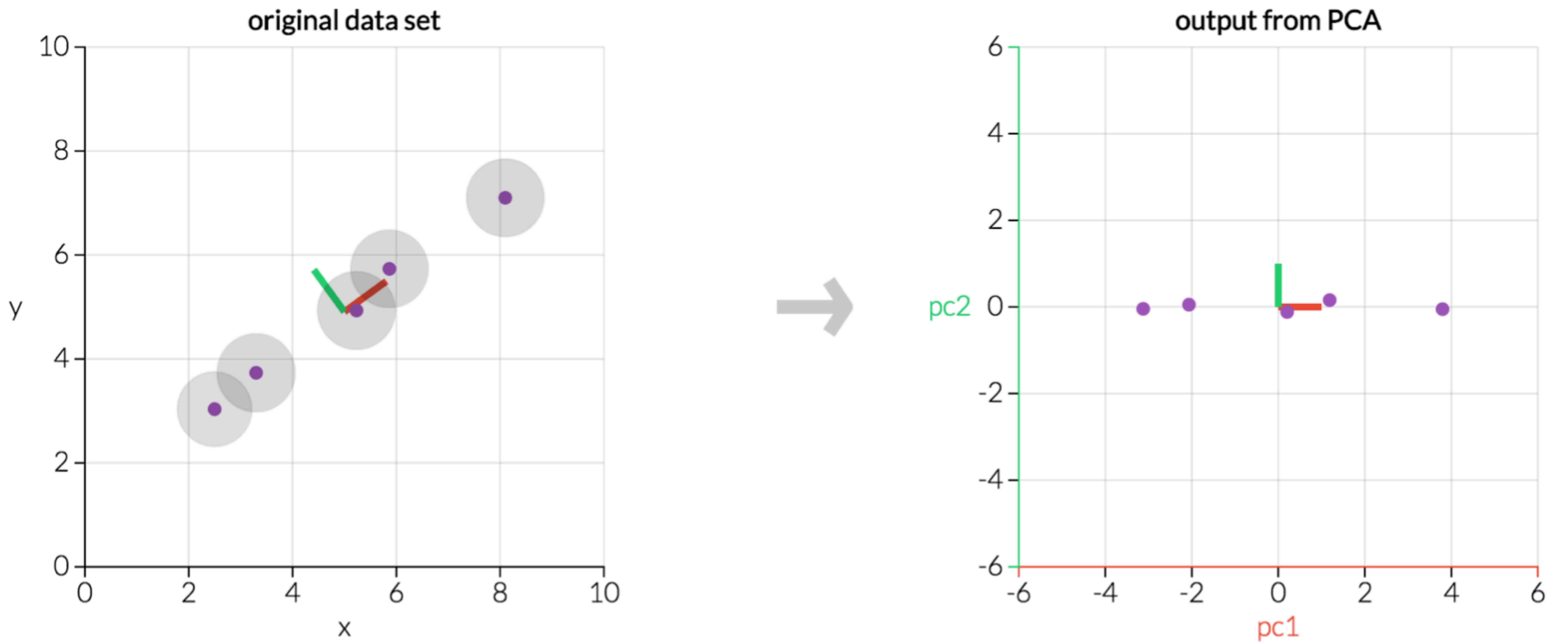
The Idea: A toy example

- They can be stored using only 9 bytes (50% savings!): Store one point (3 bytes) + the multiplying constants (6 bytes)

$$\begin{bmatrix}1\\2\\3\end{bmatrix} = 1 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}2\\4\\6\end{bmatrix} = 2 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}4\\8\\12\end{bmatrix} = 4 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\begin{bmatrix}3\\6\\9\end{bmatrix} = 3 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}5\\10\\15\end{bmatrix} = 5 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}6\\12\\18\end{bmatrix} = 6 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

PCA


original data set
10
8
6
4
2
0
y
0
2
4
6
8
10
x
output from PCA
6
4
2
0
-2
-4
-6
pc2
-6
-4
-2
0
2
4
6
pc1

High-Dimensional Mapping

- Mapping multidimensional space into space of fewer dimensions
  - typically 2D for clarity
  - 1D/3D possible?
  - keep/explain as much variance as possible
  - show underlying dataset structure
- Linear vs. non-linear approaches
  - Linear (subspace) methods
  - Non-linear dimensionality reduction

Linear vs Non-linear

- Linear Methods
    - Principal Component Analysis (PCA) – Hotelling[33]
    - Independent component analysis (ICA),
    - Linear discriminant analysis (LDA)
    - Multidimensional Scaling (MDS) – Young[38]
    - Nonnegative Matrix Factorization (NMF) – Lee[99]
- Non-linear Methods
    - Locally Linear Embeddings – Roweis[00]
    - IsoMap - Tenenbaum[00]
    - Charting – Brand[03]

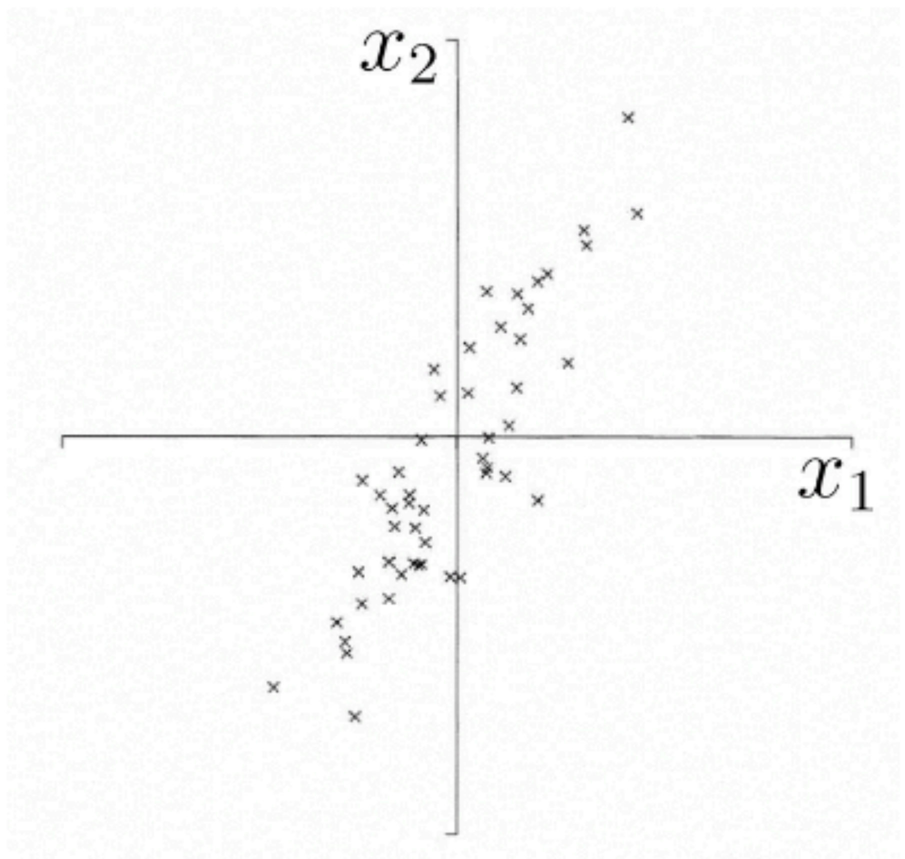
Principal Component Analysis

PCA

- Technique useful for compression and classification of data
- Find new descriptors smaller than original variables
- Retain most of sample's information - correlation between original variables
- New descriptors are principal components (PCs)
- Uncorrelated, and ordered by fraction of total information retained in each PC

Preamble

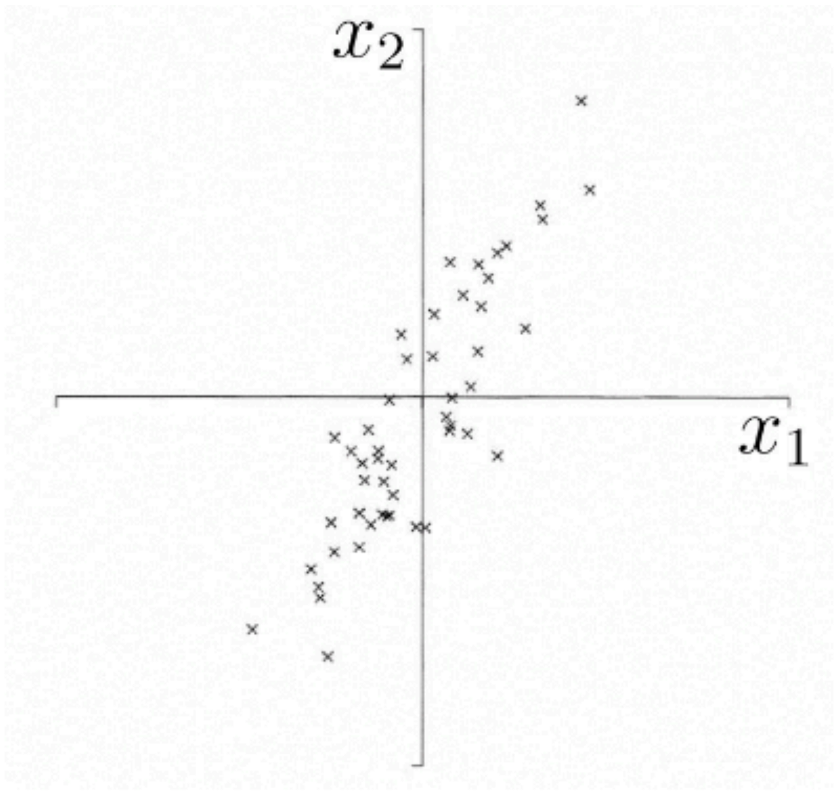
- A sample of n observations in the 2-D space
- Goal: Account for variation in a sample with as few variables as possible

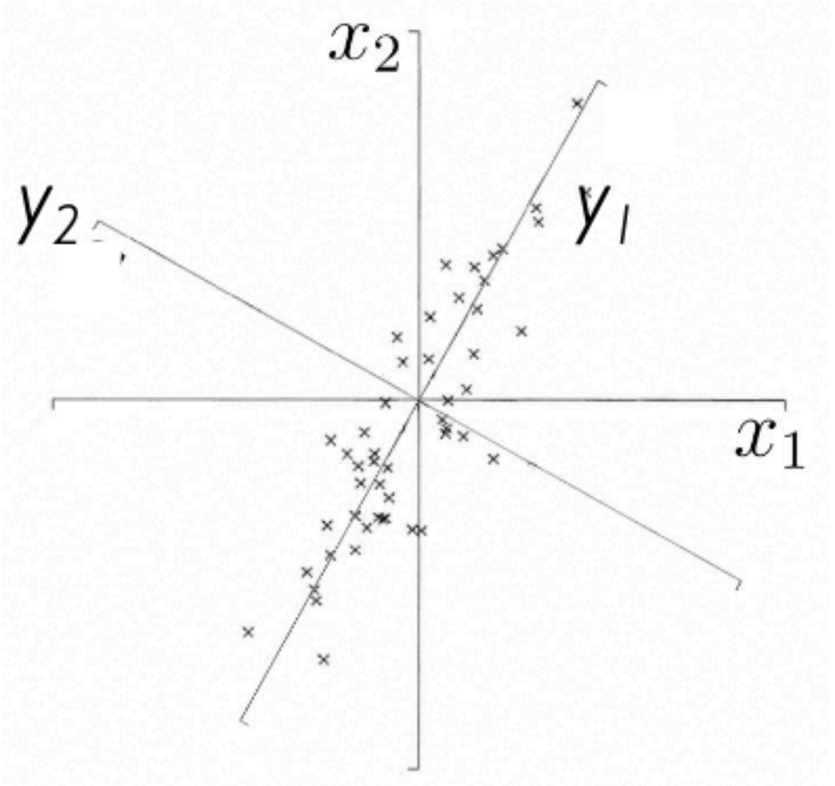


$$\mathbf{x} = (x_1, x_2)$$

Principal Components (PCs)

- Series of linear least squares fits to a sample
- Each orthogonal to all the previous
- First PC y1 is minimum distance fit to a line in space
- Second PC y2 is minimum distance fit to line in plane
- Perpendicular to first PC





PCA: General Methodology

- k original variables: $x_1, x_2, \ldots., x_k$
- Produce k new variables: $y_1, y_2, \ldots., y_k$

$$y_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1k}x_k$$

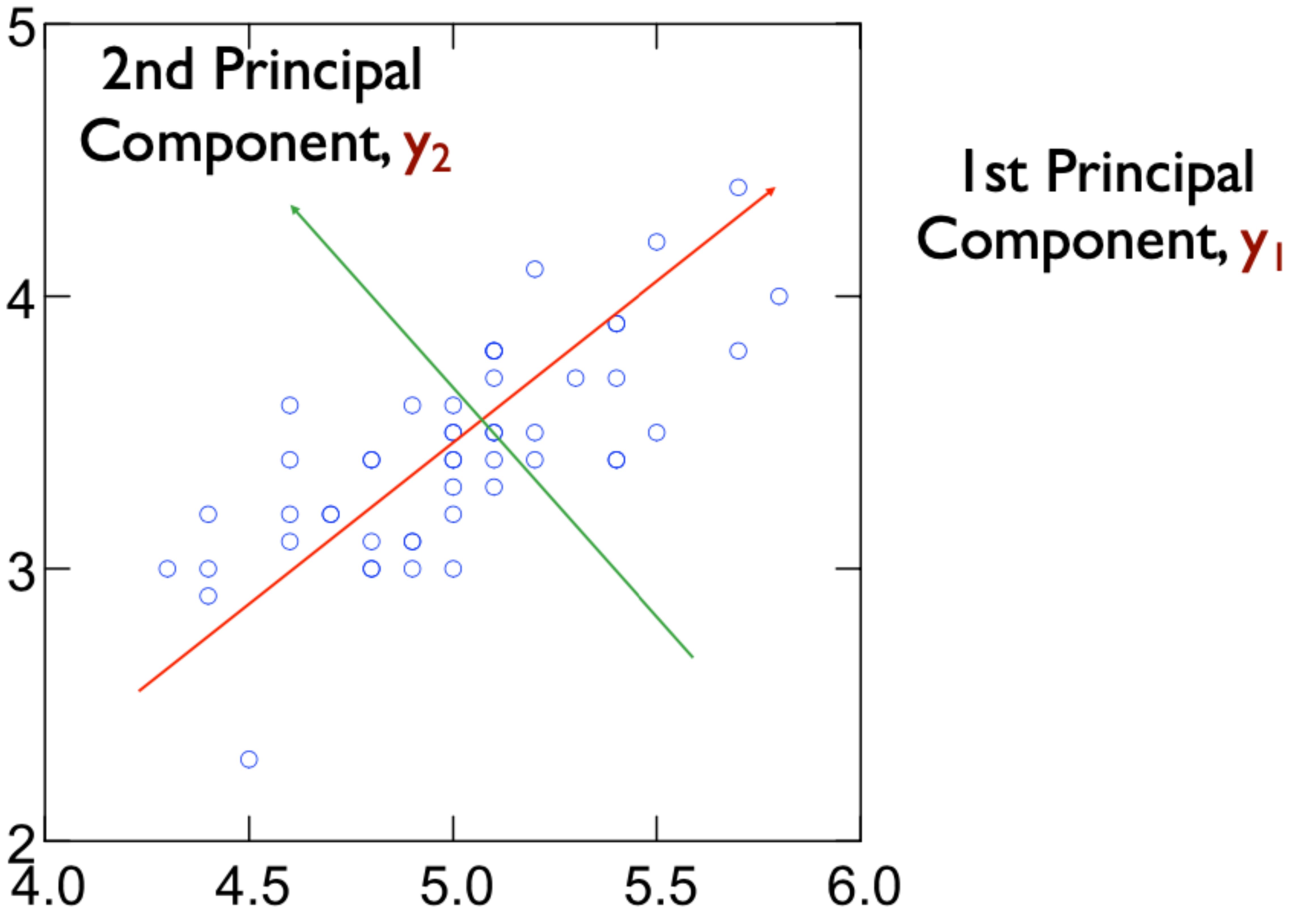
$$y_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2k}x_k$$

…

$$y_k = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kk}x_k$$

- y_k's: Principal Components
- y_k's are uncorrelated
- y_1 explains as much as possible of original variance in data set
- y_2 explains as much as possible of remaining variance

PCA



2nd Principal
Component, y2
1st Principal
Component, y1
5
4
3
2
4.0
4.5
5.0
5.5
6.0

PCA

- Uses:
  - Correlation matrix, or
  - Covariance matrix when variables in same units

PCA

- $\{a_{11}, a_{12}, ..., a_{1k}\}$ is 1^{st} Eigenvector of correlation/covariance matrix, and coefficients of first principal component
- $\{a_{21}, a_{22}, ..., a_{2k}\}$ is 2^{nd} Eigenvector of correlation/covariance matrix, and coefficients of 2^{nd} principal component
- ...
- $\{a_{k1}, a_{k2}, ..., a_{kk}\}$ is k^{th} Eigenvector of correlation/covariance matrix, and coefficients of k^{th} principal component

Variance

- Random Variable fluctuating about mean value

$$\delta x = x - \langle x \rangle$$

- Average of the square fluctuations

$$\langle (\delta x)^2 \rangle = \langle x^2 \rangle - \langle x \rangle^2$$

Covariance

- Pair of random variables, fluctuating about mean values

$$\delta x_1 = x_1 - \langle x_1 \rangle$$

$$\delta x_2 = x_2 - \langle x_2 \rangle$$

- Average of product fluctuations

$$\langle \delta x_1 \delta x_2 \rangle = \langle x_1 x_2 \rangle - \langle x_1 \rangle \langle x_2 \rangle$$

Correlation Coefficient

- $X = (x_1, x_2, x_3, \ldots, x_m)$
- $Y = (y_1, y_2, y_3, \ldots, y_m)$

Pearson's correlation coefficient: measure the linear correlation between gaussian random variables.

$$S(X,Y) = \sum_{l=1,m} \left(\frac{X_l - \overline{X}}{\Phi_X} \right) \left(\frac{Y_l - \overline{Y}}{\Phi_Y} \right)$$

$$-1 \leq S(X,Y) \leq 1$$

Covariance Matrix

$$C_{ij} = \langle x_i x_j \rangle - \langle x_i \rangle \langle x_j \rangle$$

- N random variables
- NxN symmetric matrix
- Diagonal elements are variances

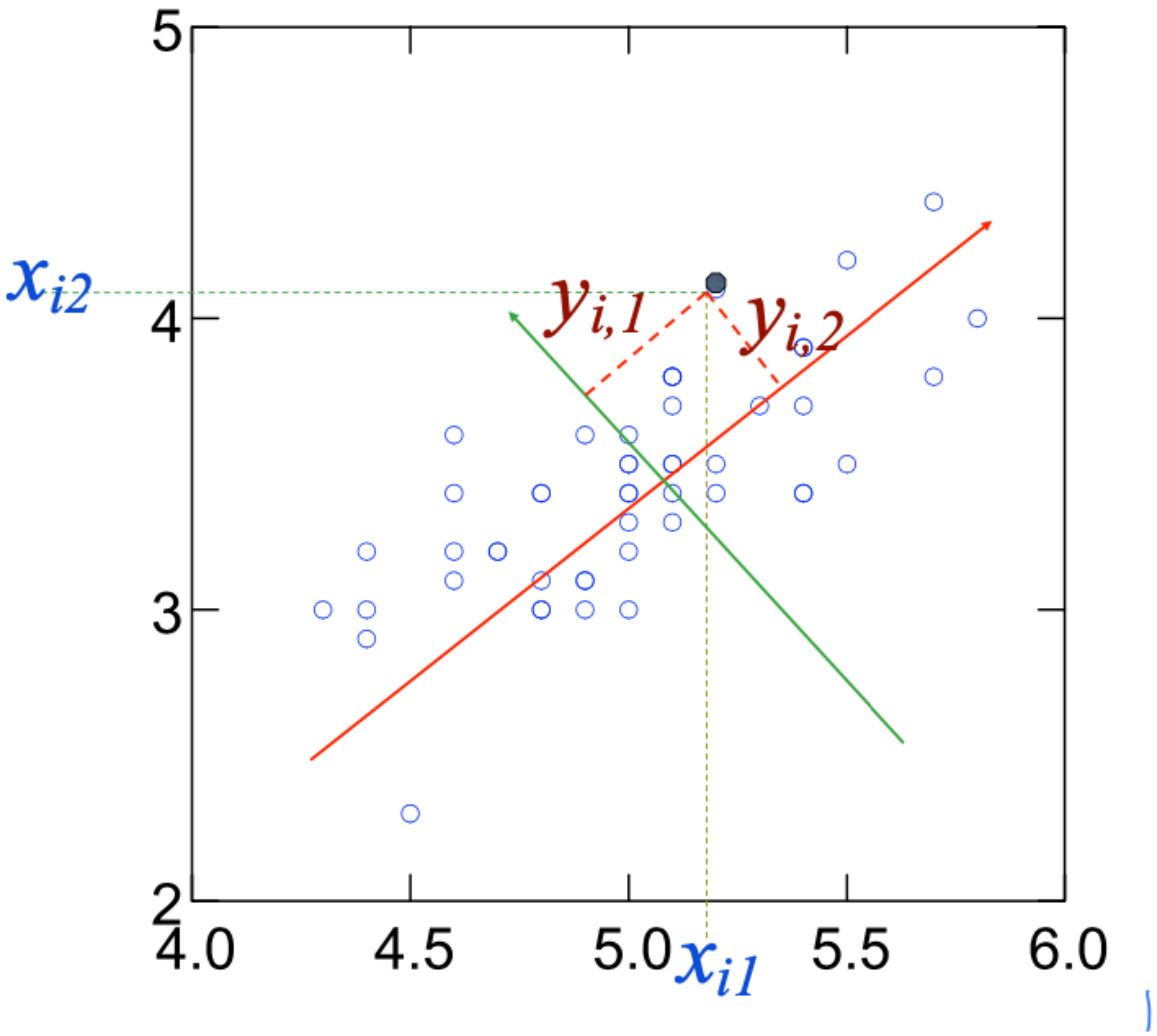
Eigen Problem

- The eigenvalue problem is any problem having the following form:

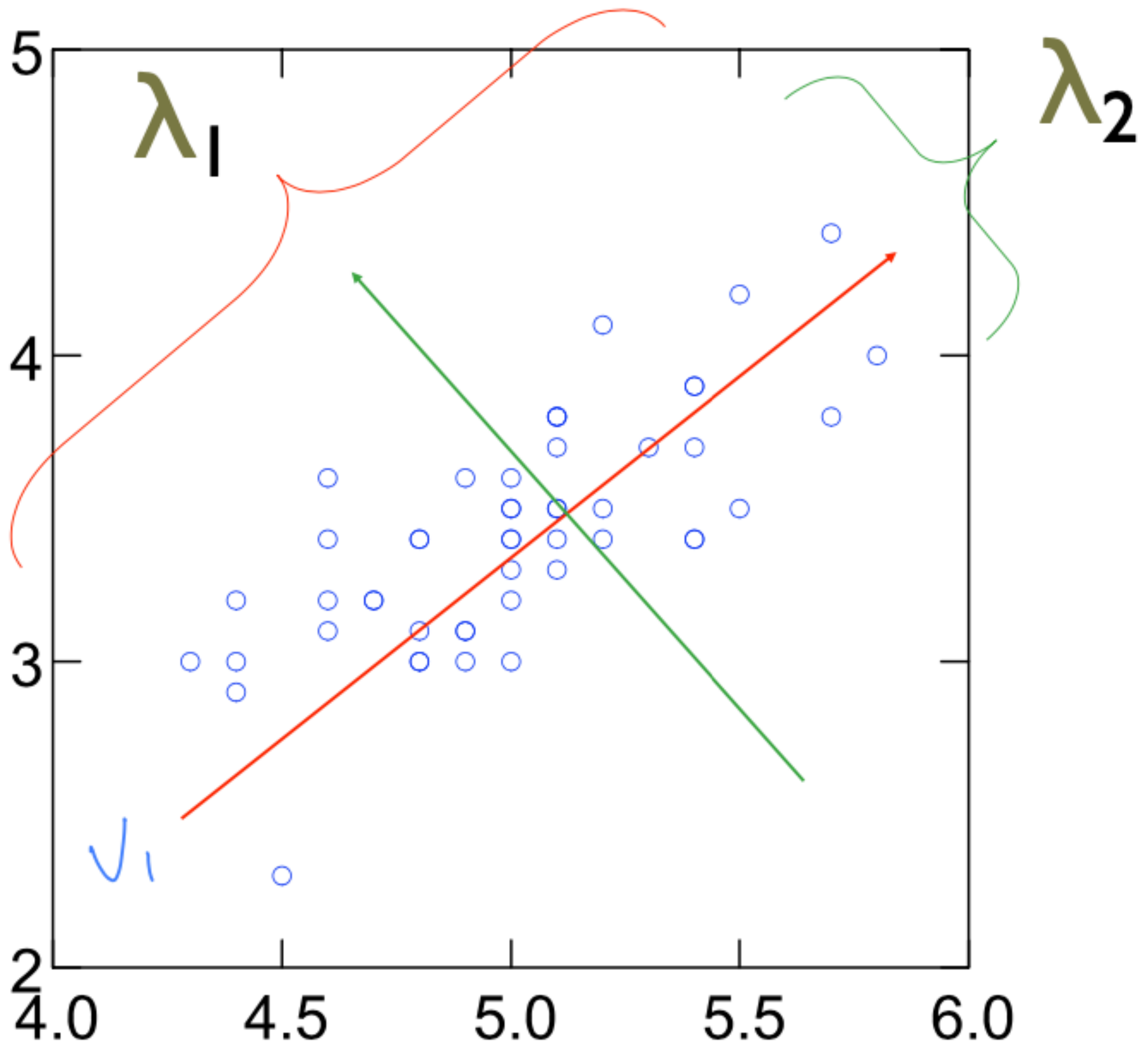
$$A.\ v = \lambda\ .v$$

- A: n x n matrix
- v: n x 1 non-zero vector
- λ: scalar
- eigenvalue of A: Value of λ satisfying equation
- eigenvector of A: Vector v which corresponding to λ

PCA Scores



Principal Eigenvalues



PCA Terminology

- *j*th **principal component** is *j*th eigenvector of covariance matrix
- **coefficients,** a_{jk}, are elements of eigenvectors and relate original variables (standardized if using correlation matrix) to components
- **amount of variance accounted for** by component is given by eigenvalue, λ_j
- **proportion of variance accounted for** by component is given by λ_j / Σ λ_j
- **loading** of *k*th original variable on *j*th component is given by $a_{jk}\sqrt{\lambda_j}$ --correlation between variable and component

PCA Mechanics

Suppose $x_1, x_2, \cdots, x_M$ are N x 1 vectors

1. Find mean
2. Subtract the mean $\Phi_i = x_i - \bar{x}$
3. Form N x M matrix $\mathbf{A} = [\Phi_1 \Phi_2 \cdots \Phi_M]$
4. Compute covariance matrix $\mathbf{C} = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T = \mathbf{A}\mathbf{A}^T$
5. Compute eigenvalues of $\mathbf{C} : \lambda_1 > \lambda_2 > \cdots \lambda_N$
6. Compute eigenvectors of $\mathbf{C} : u_1, u_2, \cdots, u_N$

PCA Mechanics

Dimensionality reduction step

$$(x - \bar{x}) = b_1 u_1 + b_2 u_2 + \cdots + b_N u_N = \sum_{i=1}^{N} b_i u_i \qquad b_i = u_i^T (x - \bar{x})$$

Keep only terms corresponding to K largest eigenvalues

$$(\hat{x} - \bar{x}) = \sum_{i=1}^{K} b_i u_i \text{ where K << N}$$

Linear Transformation

- The linear transformation $\mathcal{R}^N \rightarrow \mathcal{R}^K$ that performs the dimensionality reduction is

$$\begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_K \end{pmatrix} = \begin{pmatrix} u_1^T \\ u_2^T \\ \cdots \\ u_K^T \end{pmatrix} (x - \bar{x}) = \mathbf{U}^T (x - \bar{x})$$

- To choose K we can use the following criterion

$$\frac{\sum_{i=1}^{K} \lambda_i}{\sum_{i=1}^{N} \lambda_i} > \text{Threshold (e.g., } 0.9 \text{ or } 0.95)$$

PCA Error

- PCA preserves as much information as possible by minimizing the reconstruction error

$$e = \|x - \hat{x}\|$$

$$\hat{x} - \bar{x} = \sum_{i=1}^{K} b_i u_i$$

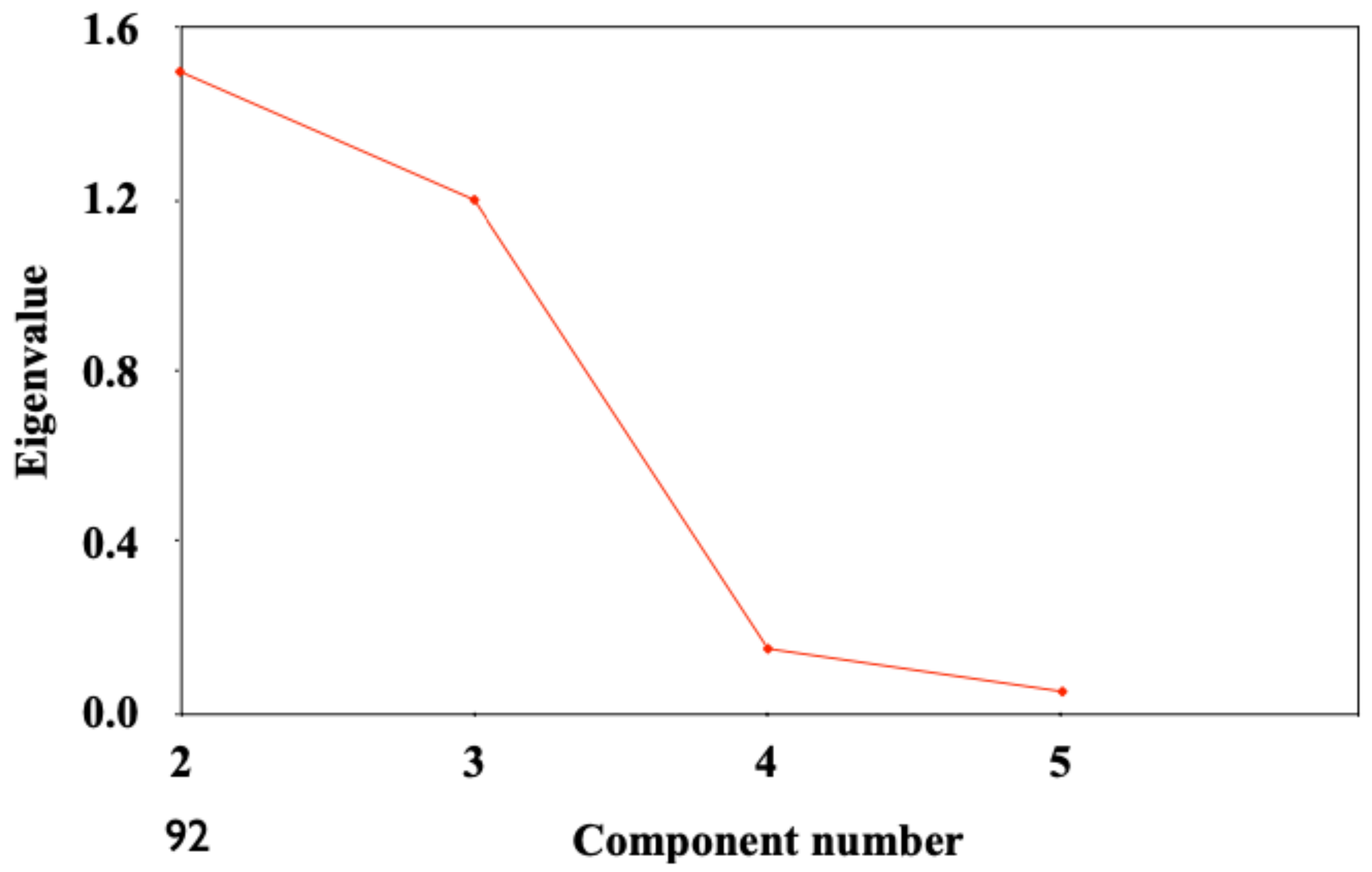
$$\hat{x} = \sum_{i=1}^{K} b_i u_i + \bar{x}$$

- The average error due to dimensionality reduction is equal to:

$$\bar{e} = \frac{1}{2} \sum_{i=K+1}^{N} \lambda_i$$

How many PCs?

- If $\lambda_j < 1$ - component explains less variance than original variable (correlation matrix)
- Use 2 components (or 3) for visual ease
- Dimensionality - number of PCs



Applications of PCA

- Facial Recognition
- Facial expression recognition
- Quantitative finance
- Medical data correlation

Summary

- Dimensionality Reduction
- PCA
- PCA Methodology
- Eigen Problem
- PCA Mechanics
- PCA Error

More Resources

- https://www.cs.mcgill.ca/~sqrt/dimr/dimreduction.html
- https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html