Lecture 13: Principal components analysis (PCA)

Lecturer: Jie Fu

High-Dimensional Data

- High-Dimensions = Lot of Features

Surveys Netflix

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

Food preference

	kale	taco bell	sashimi	pop tarts
Alice	10	1	2	7
Bob	7	2	1	10
Carolyn	2	9	7	3
Dave	3	6	10	2

- PCA: Unsupervised learning techniques to extract hidden dimensional structure from high dimensional dataset
    - Visualization
    - Efficient use of resources.
    - Statistical: lower dimension --> better generalization.
    - Further processing for other machine learning algorithm.

Motivating problem

- Friends' preferences of four different food choice.
- Dimension of data points: 4
- Number of data points: 4

Can we visualize the data in less than 4 dimension?

	kale	taco bell	sashimi	pop tarts
Alice	10	1	2	7
Bob	7	2	1	10
Carolyn	2	9	7	3
Dave	3	6	10	2

Table 1: Your friends' ratings of four different foods.

Motivating problem

- Each row of the data can be expressed approximately:

	kale	taco bell	sashimi	pop tarts
Alice	10	1	2	7
Bob	7	2	1	10
Carolyn	2	9	7	3
Dave	3	6	10	2

Table 1: Your friends' ratings of four different foods.

Name	(a_1, a_2)
Alice	$(1,\ 1)$
Bob	$(1,\ -1)$
Carolyn	$(-1,\ -1)$
Dave	$(-1,\ 1)$

Table 1: Values of (a_1, a_2) for each person

$$\bar{\mathbf{x}} + a_1\mathbf{v}_1 + a_2\mathbf{v}_2,$$

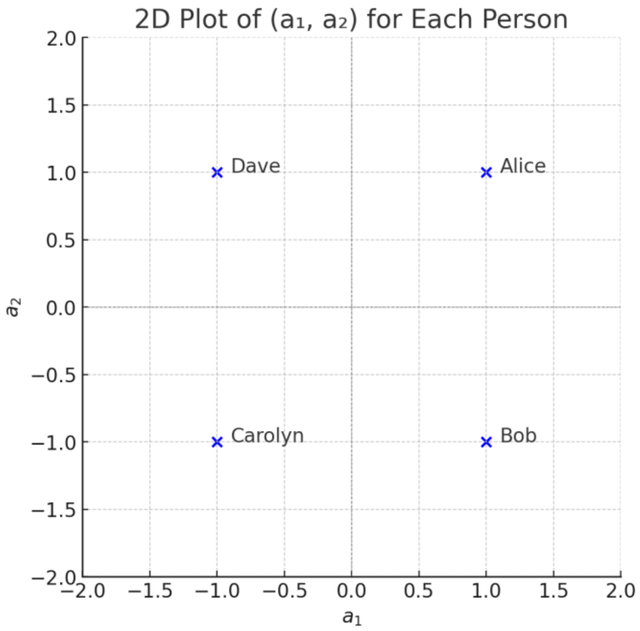
where

$$\bar{\mathbf{x}} = (5.5, 4.5, 5, 5.5)$$

is the average of the data points,

$$\mathbf{v}_1 = (3, -3, -3, 3),$$

$$\mathbf{v}_2 = (1, -1, 1, -1),$$



The role of PCA

- Reduce the dimensionality of data points (eg. 4 to 2):
- Given a list of m n-dimensional vectors (data points),

$$x_1, x_2, \ldots, x_m \in R^n$$

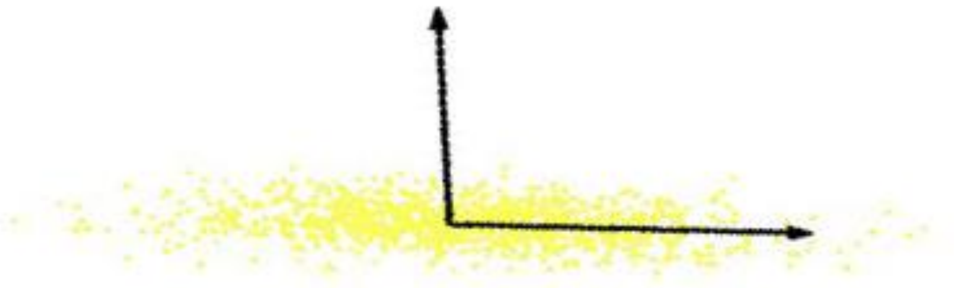
For each vector x_i, express it as linear combinations of k n-dimensional vectors $v_1, \ldots, v_n \in R^n$ such that

$$x_i \approx \sum_{j \in 1}^{k} a_{ij} v_j$$

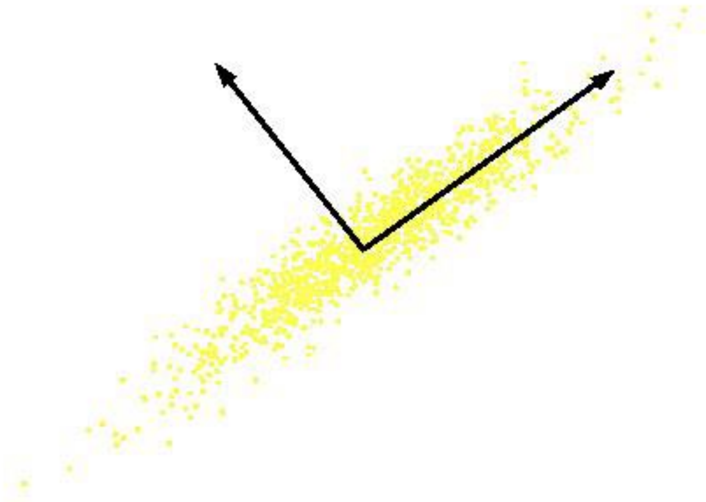
Dimension reduction: n → k, which is smaller than n.

PCA

- PCA is an orthogonal projection or transformation of the data into a possible lower dimensional subspace so that **the variance of the projected data is maximized.**

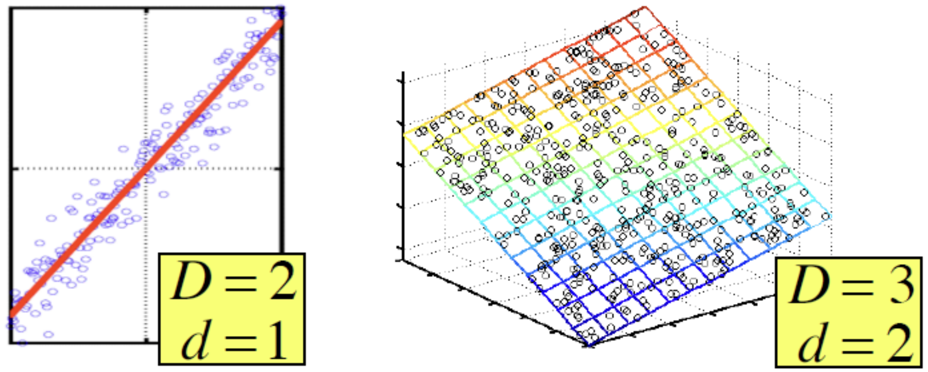


Only one relevant feature



Both features are relevant, but

PCA

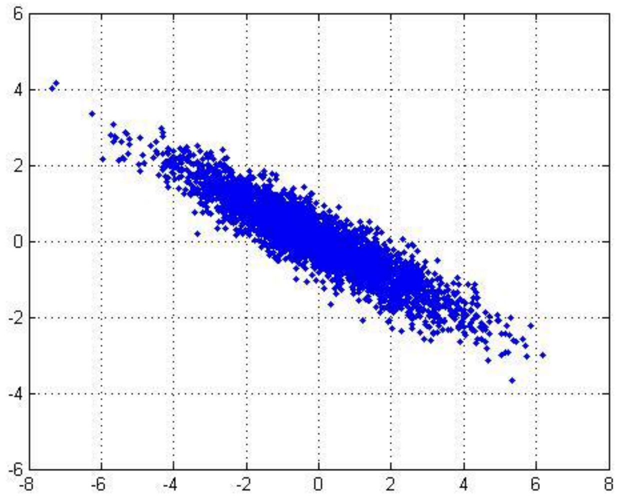


Does the data mostly lie in a subspace?
If so, what is its dimensionality?

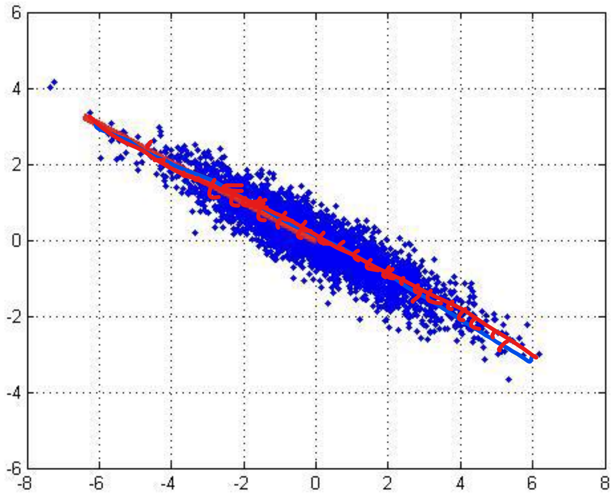
- The goal is to identify the axes or subspace the high-dimensional data should be projected into.

Maximize the variance

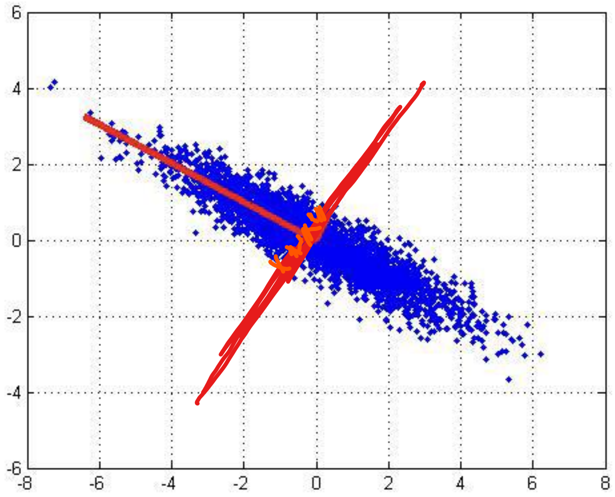
- Why maximize the variance of the projected data?



2D Gaussian dataset



1st principle component

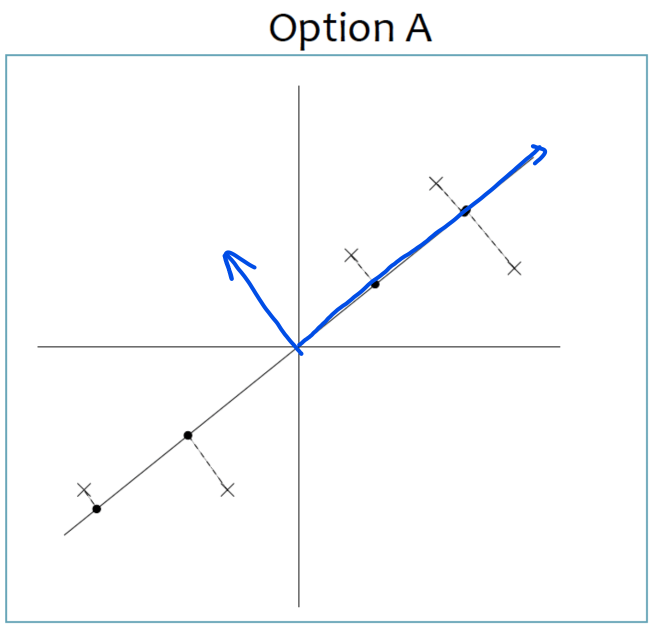


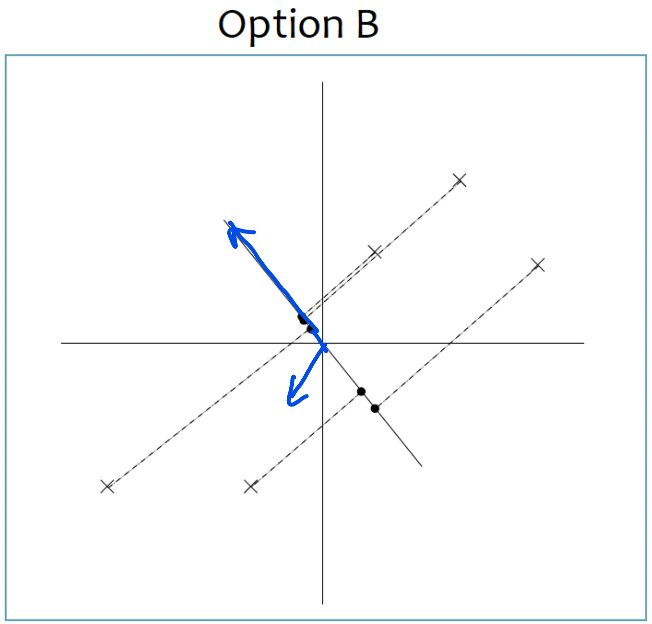
2nd principle component

Variance tells us how much information or “spread” a dataset has. In PCA, we assume directions with higher variance are more informative.

Maximize the variance

- Which of the two projections maximize the variance?





Figures from Andrew Ng
(CS229 Lecture Notes)

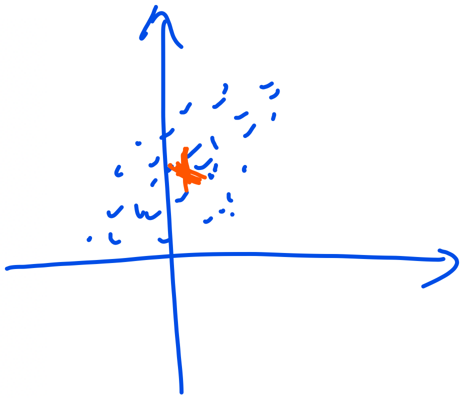
Maximize the variance

We want to find new axes (directions) to project our data such that:

- The projected data has **maximum variance**.
- The new features (called principal components) are **uncorrelated**.

	kale	taco bell	sashimi	pop tarts
Alice	10	1	2	7
Bob	7	2	1	10
Carolyn	2	9	7	3
Dave	3	6	10	2

Table 1: Your friends' ratings of four different foods.



Step 1: center the data matrix
Step 2: compute the covariance matrix of the centered data
Step 3: select top k principal components/features

Step 1 and step 2

- Center the data

$$X_c = X - \bar{X}$$

- Example:

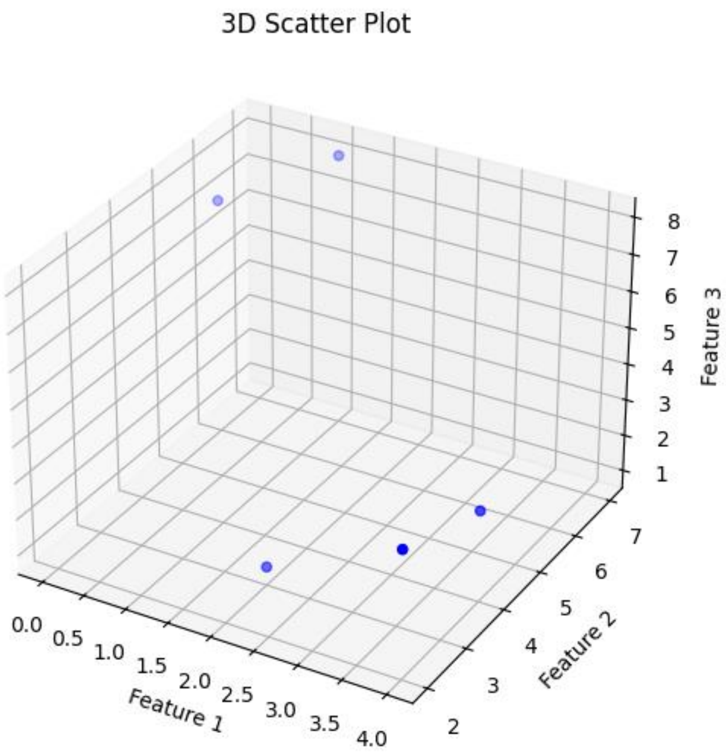
Consider the following dataset with 5 samples and 3 features:

$$\begin{array}{cc} & \begin{array}{ccc} \text{feature 1} & 2 & 3 \end{array} \\ X = & \begin{bmatrix} 2 & 3 & 1 \\ 4 & 2 & 4 \\ 4 & 4 & 3 \\ 0 & 6 & 7 \\ 1 & 7 & 8 \end{bmatrix} \end{array}$$

mean 1   2   3

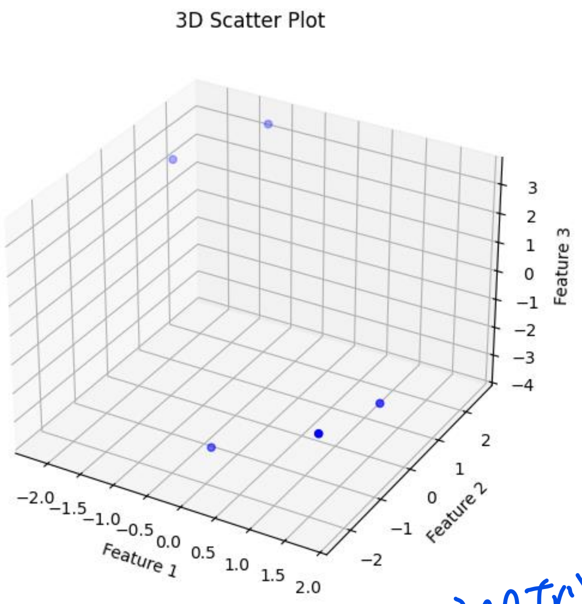
2.2   4.4   4.6

Mean of each feature



- Centered data matrix

$$X_c = \begin{bmatrix} -0.2 & -1.4 & -3.6 \\ 1.8 & -2.4 & -0.6 \\ 1.8 & -0.4 & -1.6 \\ -2.2 & 1.6 & 2.4 \\ -1.2 & 2.6 & 3.4 \end{bmatrix}$$



$K = U \Lambda U^{-1}$

Covariance matrix after rotation $\leftarrow \Lambda \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \ddots \lambda_n \end{bmatrix}$

Step 2: compute the covariance matrix of the centered data (use the transposed.)

np.cov(M_c.T)

The covariance matrix K is given by:

Δ

$$K = \frac{1}{n-1} X_c^\top X_c$$

$$\begin{array}{cc} & \begin{array}{ccc} \mathrm{var}(X_i) & \mathrm{cov}(X_1, X_2) & \end{array} \\ \mathrm{Cov}(X_2, X_1) & \begin{bmatrix} 3.2 & -2.85 & -3.15 \\ -2.85 & 4.3 & 4.95 \\ -3.15 & 4.95 & 8.3 \end{bmatrix} \end{array}$$

Eigenvalue and eigenvectors of a matrix

UNIVERSITY of FLORIDA

Let A be a $n \times n$ matrix.

- $\vec{x} \neq 0$ is an *eigenvector* of A if there is a scalar λ such that

$$A\vec{x} = \lambda\vec{x}$$

- the corresponding λ is called the *eigenvalue*.

- Example: find the eigenvalue and eigenvector of A.

$$A = \begin{bmatrix} 4 & 2 \\ 1 & 3 \end{bmatrix}$$

$$A = \begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix} \quad ?$$

$$A\vec{x} = \lambda \cdot \vec{x}$$

$$\begin{bmatrix} 5 & 0 \\ 0 & 10 \end{bmatrix} \cdot \begin{bmatrix} c \\ 0 \end{bmatrix} = \underline{5} \cdot \begin{bmatrix} c \\ 0 \end{bmatrix}$$

$$\begin{cases} 5x_1 + 0 \cdot x_2 = \lambda x_1 \\ 0 \cdot x_1 + 10 \cdot x_2 = \lambda x_2 \end{cases}$$

$\lambda = 5$: $10x_2 = 5x_2$

$\Downarrow$

$x_2 = 0$

$\lambda = 10$: $x_1 = 0$

$$\begin{bmatrix} 0 \\ c_2 \end{bmatrix}$$

Eigenvalue and eigenvectors of a matrix

$$Ax = \lambda x \Rightarrow (A - \lambda I)x = 0$$

λ be selected such that

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

Eigenvalue and eigenvectors of a matrix

Diagonalizable Matrices

A $n \times n$ matrix with n linearly independent eigenvectors is said to be **diagonalizable.**

$$\boldsymbol{A}\,\boldsymbol{u_1} = \lambda_1\,\boldsymbol{u_1},$$
$$\boldsymbol{A}\,\boldsymbol{u_2} = \lambda_2\,\boldsymbol{u_2},$$
$$\ldots$$
$$\boldsymbol{A}\,\boldsymbol{u_n} = \lambda_n\,\boldsymbol{u_n},$$

$$\Big[\,[\;]\;[\;]\;\cdots\;[\;]\,\Big] \qquad u_1 \quad u_2 \quad u_n$$

In matrix form:

$$\boldsymbol{A}\,\underbrace{(\boldsymbol{u_1}\quad \ldots \quad \boldsymbol{u_n})}_{}^{U} = (\lambda_1\boldsymbol{u_1}\quad \ldots \quad \lambda_n\boldsymbol{u_n}) = (\boldsymbol{u_1}\quad \ldots \quad \boldsymbol{u_n})\underbrace{\begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{pmatrix}}_{D}$$

This corresponds to a similarity transformation

$$\boldsymbol{AU} = \boldsymbol{UD} \iff \underset{\Delta}{\boldsymbol{A}} = \underline{\boldsymbol{UDU^{-1}}}$$

PCA and eigen-decomposition of covariance matrix.

- Covariance matrix:

$$K[i,j] = \mathrm{cov}(\underbrace{X_i - \bar{X}_i}_{\text{mean } i},\ \underbrace{X_j - \bar{X}_j}_{\text{mean } j})$$

$$K = U \underbrace{\Lambda}_{\text{diagonal matrix.}} U^T$$

Property of covariance matrix:

1. It is symmetric → for symmetric matrix, eigenvectors for distinct eigenvalues are orthogonal.

   any i, j $\quad v_i, v_j$ are orthogonal.

   $U = [v_1\ v_2 \cdots v_n]$

   $\theta = \pm \frac{\pi}{2}$: $v_i \cdot v_j = v_i^T v_j = 0$

2. It is real: -> All eigenvalues of a real symmetric matrix are real.

   linalg.eig(A) $\qquad$ orthonormal vectors

   $v_i, v_j \quad \|v_i\| = 1$

   eg. $v_1 = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad v_2 = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$

   $\downarrow \frac{v}{\|v\|}$

   $v_1' = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad v_2' = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

PCA and eigen-decomposition of covariance matrix.

- Eigen-decomposition of covariance matrix

$$K = U\Lambda U^{-1}$$

diagonal

cov after coordinate transformation

$$\begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}$$

$$K = U\Lambda U^{-1}$$

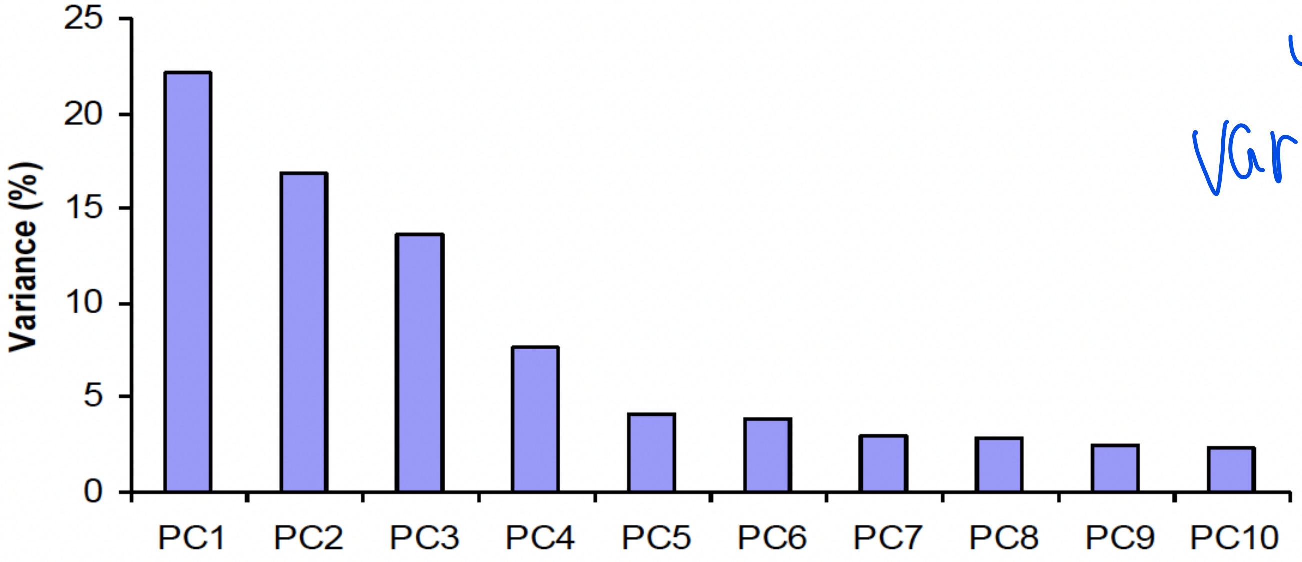
- Columns of U are *eigenvectors* of K.
- Diagonal matrix Λ are eigenvalues of K, ordered in the order of eigenvectors.

$$\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$$

- We order these eigenvectors in an order of the values of eigenvalues and called these: 1st principal component, 2nd principal component, etc.
- Where does dimensionality reduction come from?
  - Can ignore the components of lesser significance.

$\lambda_1 \dots \lambda_n$

Variance of features after transform



Example

The covariance matrix K is given by:

$X_c = X - \bar{X}$ $\rightarrow$ $\mathbb{E}(X_c) = 0$

X_c: f_1 f_2 f_3

1
2
3
4
5

$$K = \frac{1}{n-1} X_c^\top X_c$$

$$\begin{bmatrix} 3.2 & -2.85 & -3.15 \\ -2.85 & 4.3 & 4.95 \\ -3.15 & 4.95 & 8.3 \end{bmatrix}$$

eigenvalues, eigenvectors = LA.eig(K)

$$\begin{bmatrix} 13.38070762 & 1.82004592 & 0.59924646 \end{bmatrix}$$

$$\begin{bmatrix} -0.38263617 & 0.77297413 & -0.50606379 \\ 0.53188845 & -0.26357343 & -0.80475072 \\ 0.75543646 & 0.57709622 & 0.31028329 \end{bmatrix}$$

PC

- Project the Data onto the Principal Components:

$$\text{proj}_v X = \frac{X \cdot V}{\|V\|} \cdot \vec{v} = (X \cdot V)\frac{\vec{V}}{\|V\|}$$

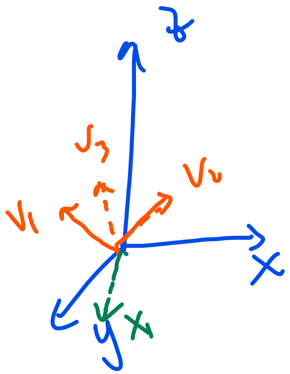
normalization

- If we want 2D dimension, project each centered data point into the first two pc:

$$\text{proj}_{v_1} V_2 = \frac{V_2 \cdot V_1}{\|V_1\|} \cdot \vec{V}_1$$

$$X_c = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{bmatrix} -0.2 & -1.4 & -3.6 \\ 1.8 & -2.4 & -0.6 \\ 1.8 & -0.4 & -1.6 \\ -2.2 & 1.6 & 2.4 \\ -1.2 & 2.6 & 3.4 \end{bmatrix}$$

$$\begin{array}{c} \begin{bmatrix} -0.38263617 & 0.77297413 & -0.50606379 \\ 0.53188845 & -0.26357343 & -0.80475072 \\ 0.75543646 & 0.57709622 & 0.31028329 \end{bmatrix} \\ \begin{array}{ccc} \quad V_1 \quad\quad\quad\quad & \quad\quad V_2 \quad\quad\quad\quad & \quad\quad V_3 \end{array} \end{array}$$



$$X_1 = a_1 V_1 + a_2 V_2 + a_3 V_3$$

$$\text{proj}_{V_1} \vec{X}_1 = \text{proj}_{V_1}(a_1 V_1 + a_2 V_2 + a_3 V_3) = a_1 \,\text{proj}_{V_1} V_1 + a_2 \,\text{proj}_{V_1} V_2 + a_3 \,\text{proj}_{V_1} V_3$$

$$= a_1 \cdot \underbrace{(V_1 \cdot V_1)}_{1} \cdot \frac{\vec{V}_1}{\cancel{\|V_1\|}} + a_2 \cdot 0 + a_3 \cdot 0 = a_1 \vec{V}_1$$

$$a_i \vec{v_i} = \mathrm{proj}_{v_i} x_k$$

v_i : eigenvector (normalized)
orthogonal.

$$X = \sum_{i=1}^{n} a_i \vec{v_i}$$

n - features

$$\Downarrow$$

$$\tilde{X} = \sum_{i=1}^{K} a_i \vec{v_i}$$

$k < n$

$$X - \tilde{X} = \sum_{i=k+1}^{n} a_i \vec{v_i}$$