


# Dimension Reduction

Mahdi Roozbahani  
Georgia Tech

# Outline

- Overview 
- Principle Component Analysis: Main Idea *PCA*
- The PCA Algorithm
- PCA and SVD
- Summary

# Motivating Example: Data Visualization

53 blood and urine samples  
(features) from 65 people

- Matrix format (65x53)

	H-WBC	H-RBC	H-Hgb	H-Hct	H-MCV	H-MCH	H-MCHC
A1	8.0000	4.8200	14.1000	41.0000	85.0000	29.0000	34.0000
A2	7.3000	5.0200	14.7000	43.0000	86.0000	29.0000	34.0000
A3	4.3000	4.4800	14.1000	41.0000	91.0000	32.0000	35.0000
A4	7.5000	4.4700	14.9000	45.0000	101.0000	33.0000	33.0000
A5	7.3000	5.5200	15.4000	46.0000	84.0000	28.0000	33.0000
A6	6.9000	4.8600	16.0000	47.0000	97.0000	33.0000	34.0000
A7	7.8000	4.6800	14.7000	43.0000	92.0000	31.0000	34.0000
A8	8.6000	4.8200	15.8000	42.0000	88.0000	33.0000	37.0000
A9	5.1000	4.7100	14.0000	43.0000	92.0000	30.0000	32.0000

**Difficult to see the correlations of different features**

# Motivating Example: Data Visualization

Is there a representation better than the coordinate axes?

Is it really necessary to show all the 53 dimensions?

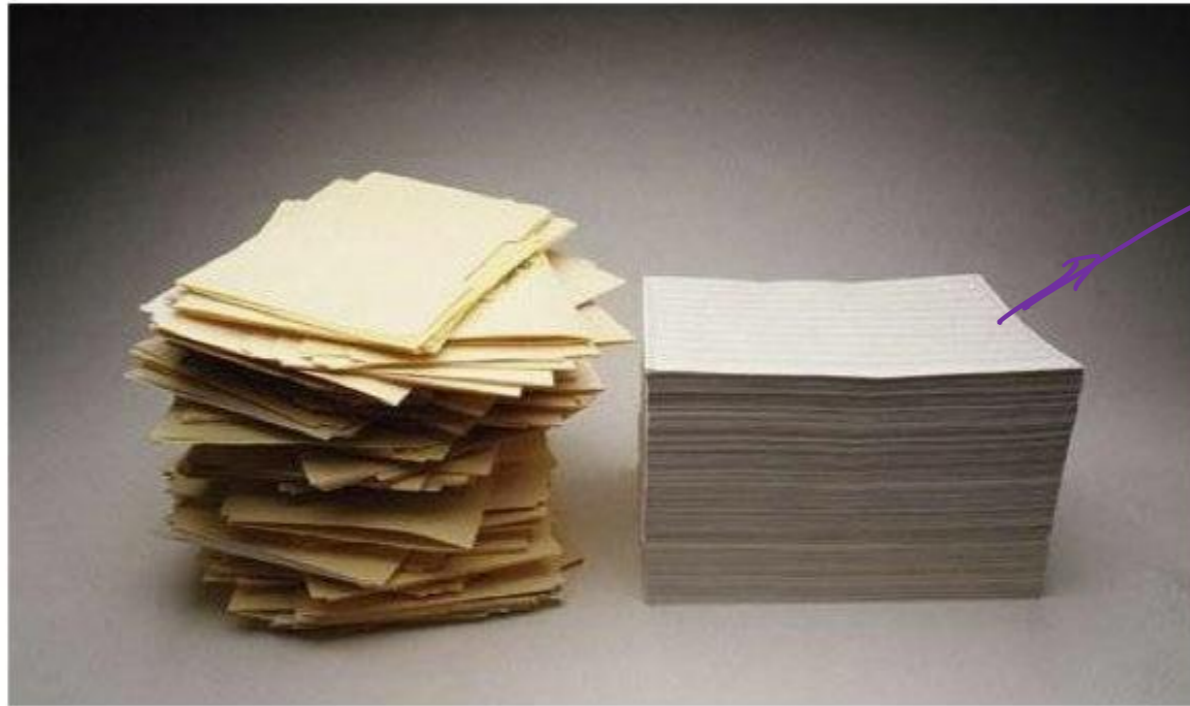
- ... what if there are strong correlations between the features?

How could we find

the *smallest* subspace of the 53-D space that keeps the *most information* about the original data?

A Solution: Dimension Reduction

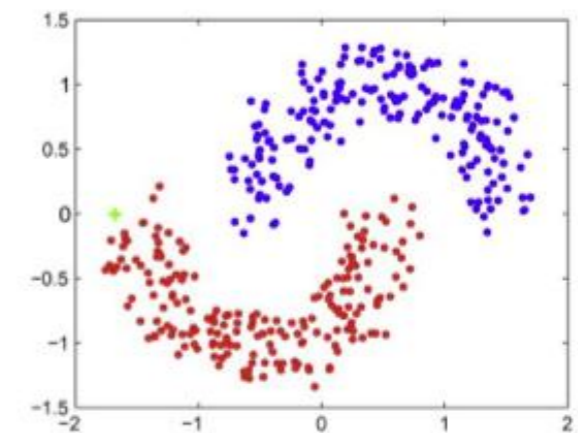
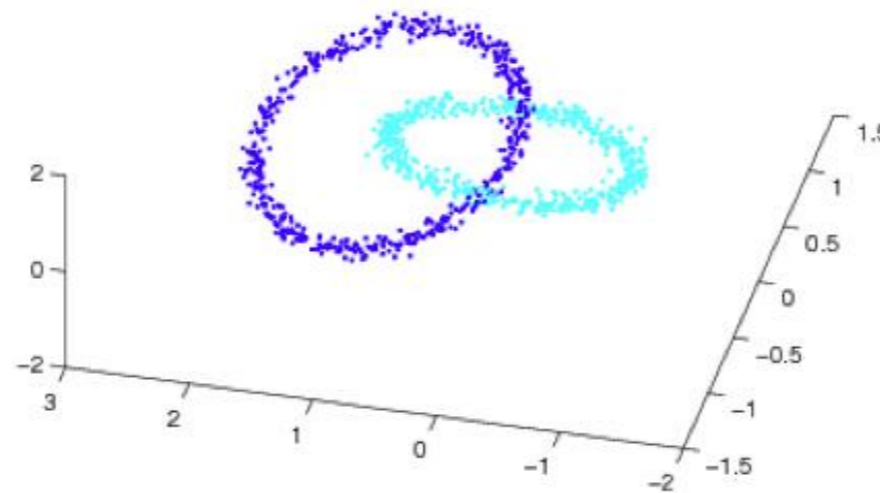
# Another Example: Dimension Reduction for Text



datapoint

Bow

What are the relations between data points?





## Term-Document Data Matrix – Bag-of-words

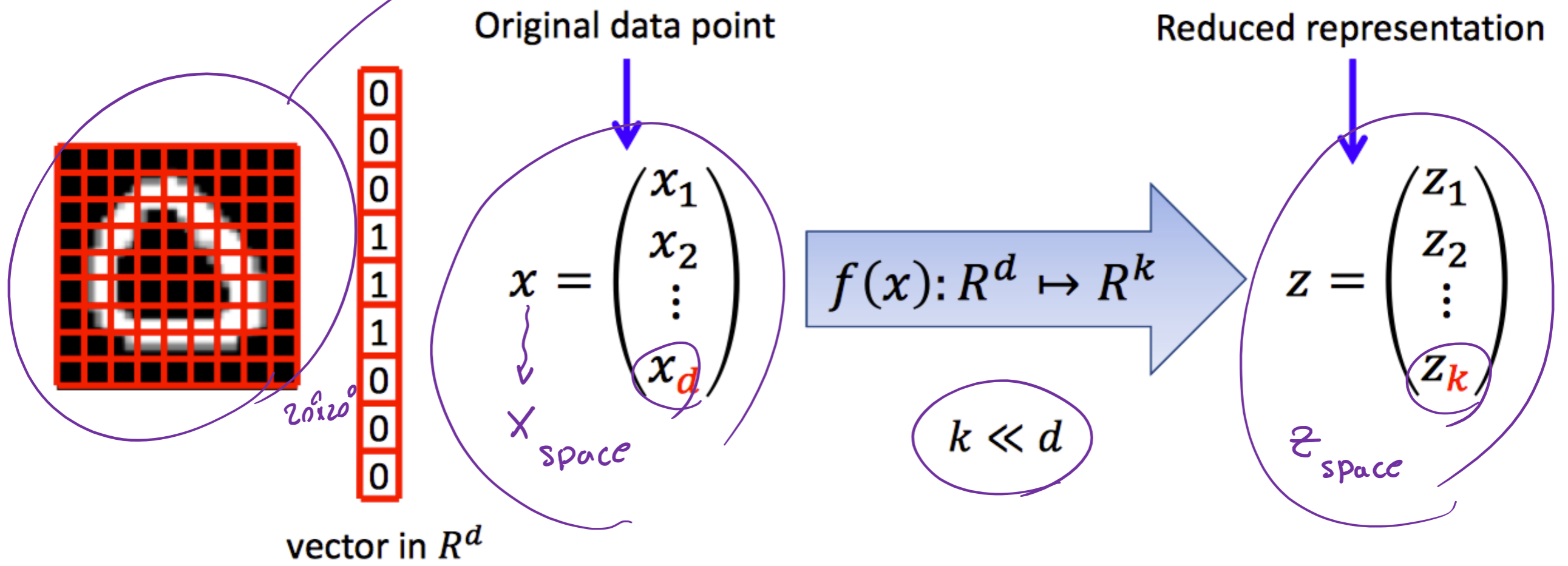
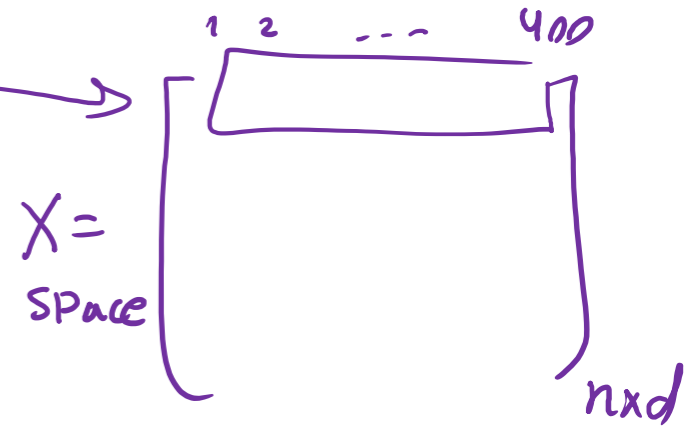
	database	SQL	index	regression	likelihood	linear
d1	24	21	9	0	0	3
d2	32	10	5	0	3	0
d3	12	16	5	0	0	0
d4	6	7	2	0	0	0
d5	43	31	20	0	3	0
d6	2	0	0	18	7	16
d7	0	0	1	32	12	0
d8	3	0	0	22	4	2
d9	1	0	0	34	27	25
d10	6	0	0	17	4	23

▪▪▪ Many more features

**Solution:  
Dimension  
Reduction**

# What is Dimension Reduction?


- The process of reducing the number of random variables under consideration
  - One can combine, transform or select variables
  - One can use linear or nonlinear operations



# Applications of Dimension Reduction

- The dimension-reduced data can be used for
  - Visualizing, exploring and understanding the data
  - Aggregating weak signals in the data
  - Cleaning the data
  - Speeding up subsequent learning task
  - Building simpler model later
- Key questions of a dimensionality reduction algorithm
  - What is the criterion for carrying out the reduction process?
  - What are the algorithm steps?

# Outline

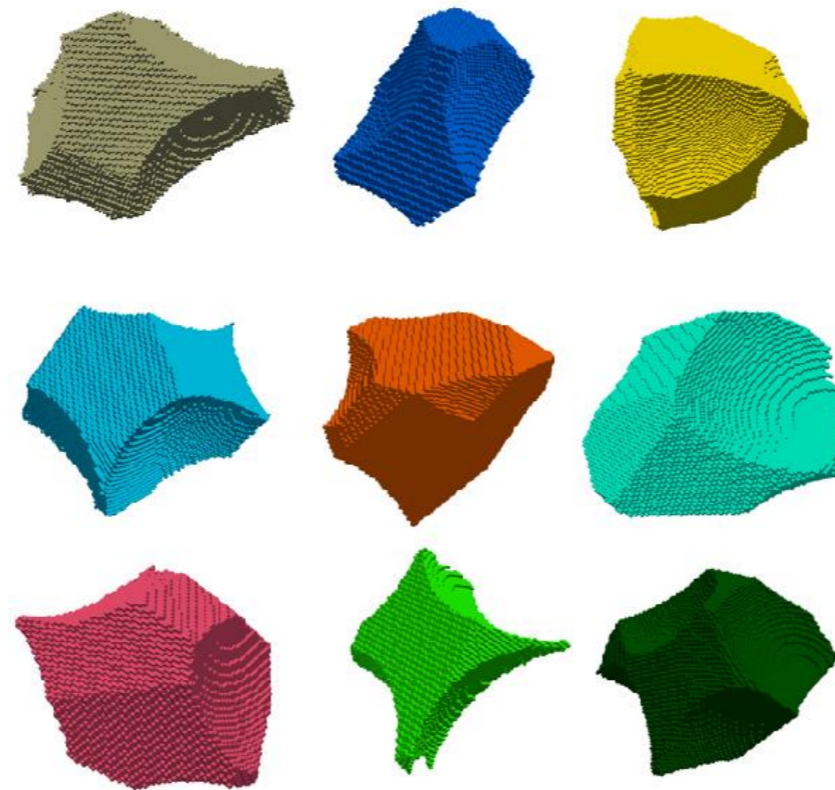
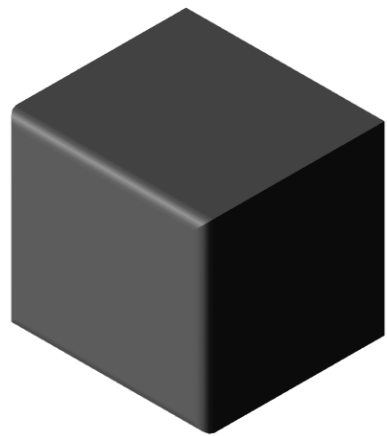
- Overview
- Principle Component Analysis: Main Idea 
- The PCA Algorithm
- PCA and SVD
- Summary

# Mahdi's example

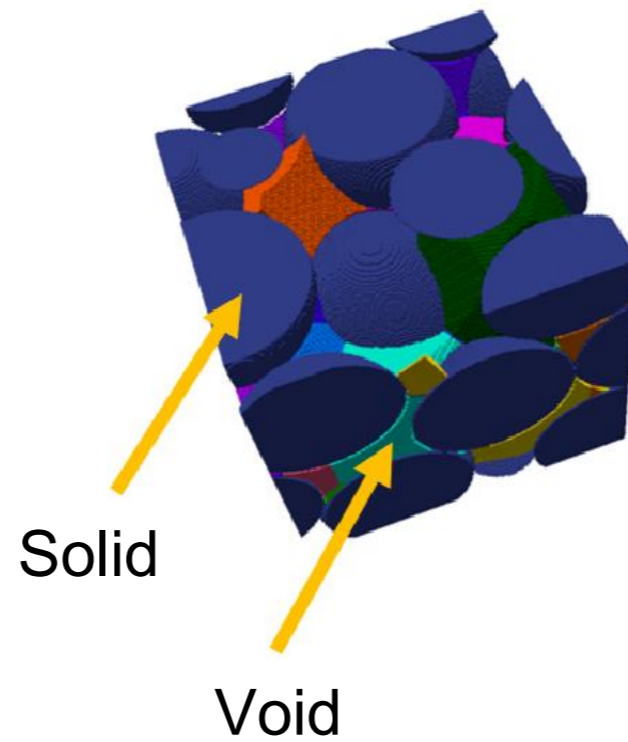
Pixel in 2D



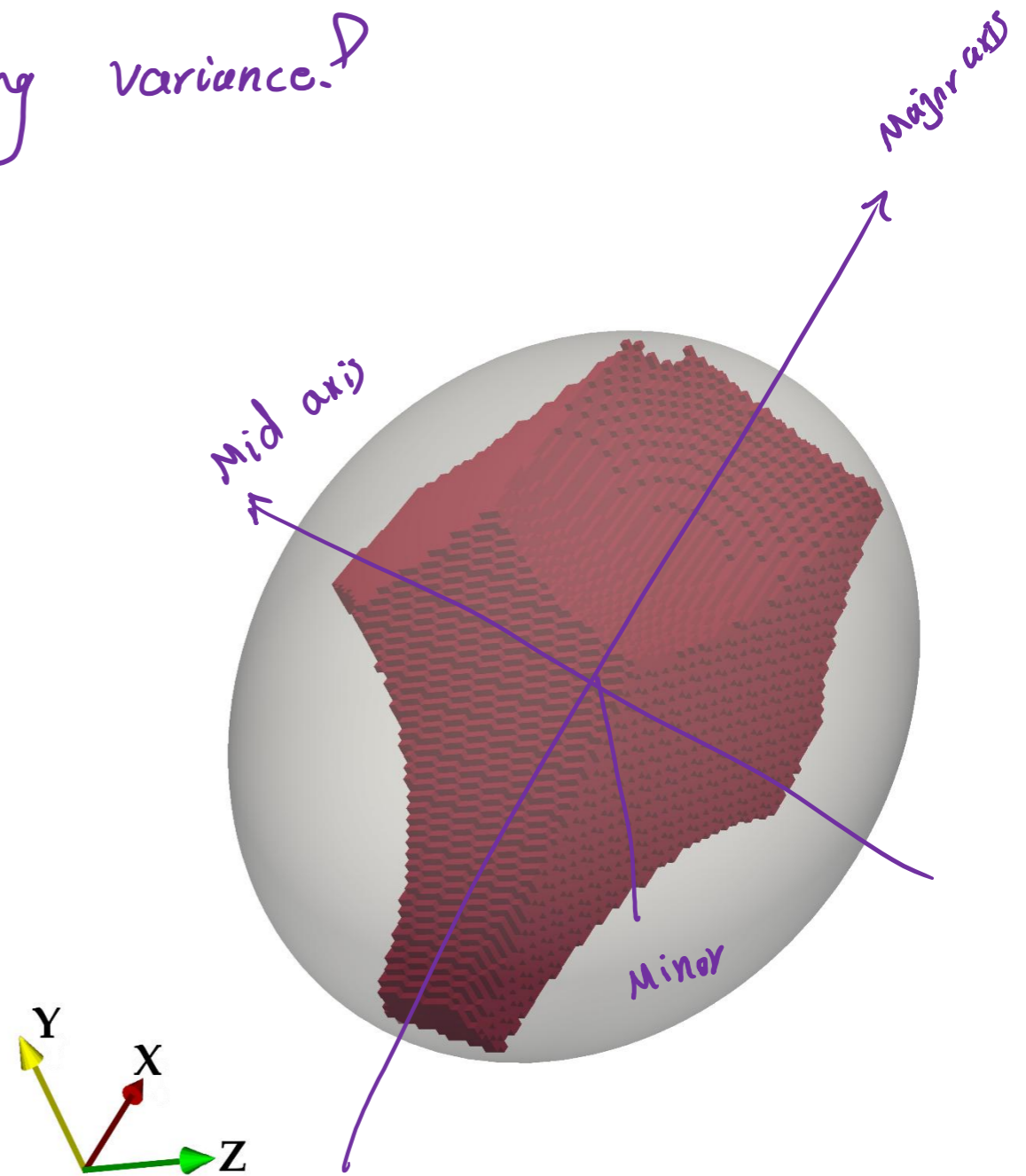
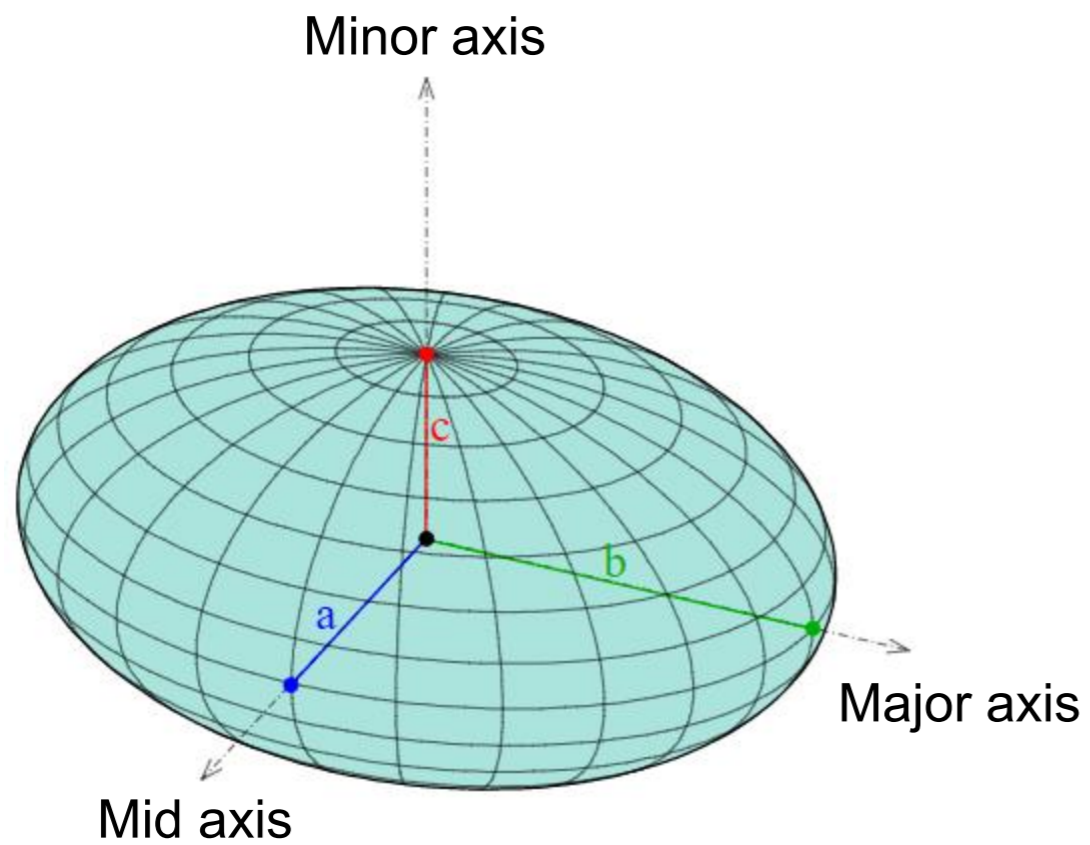
Voxel in 3D



Segmented Voids



PCA? do you mean maximizing variance?



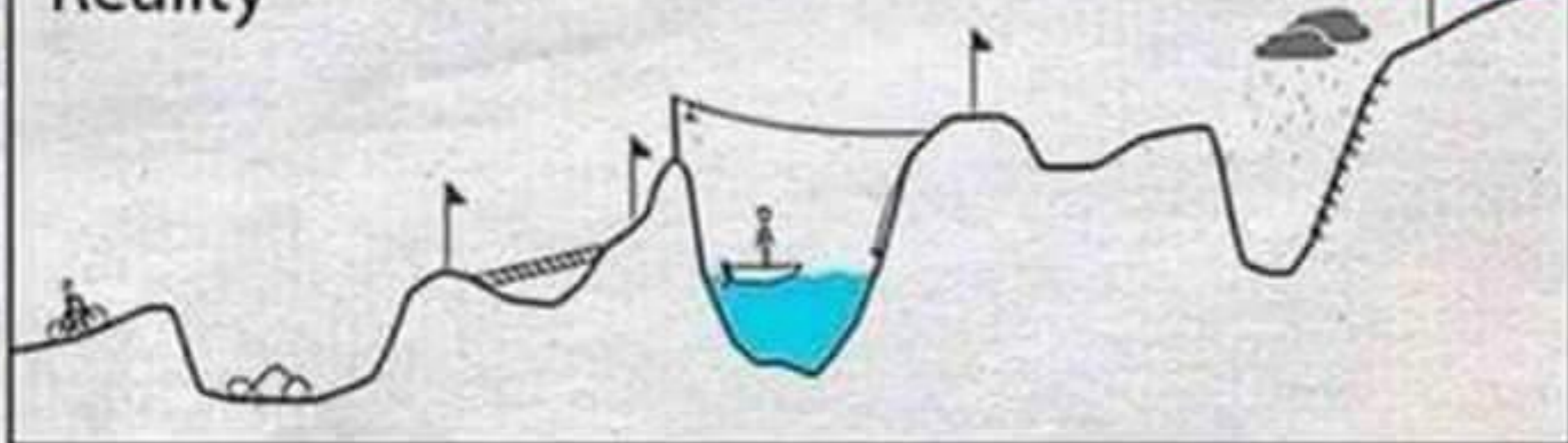
Your plan

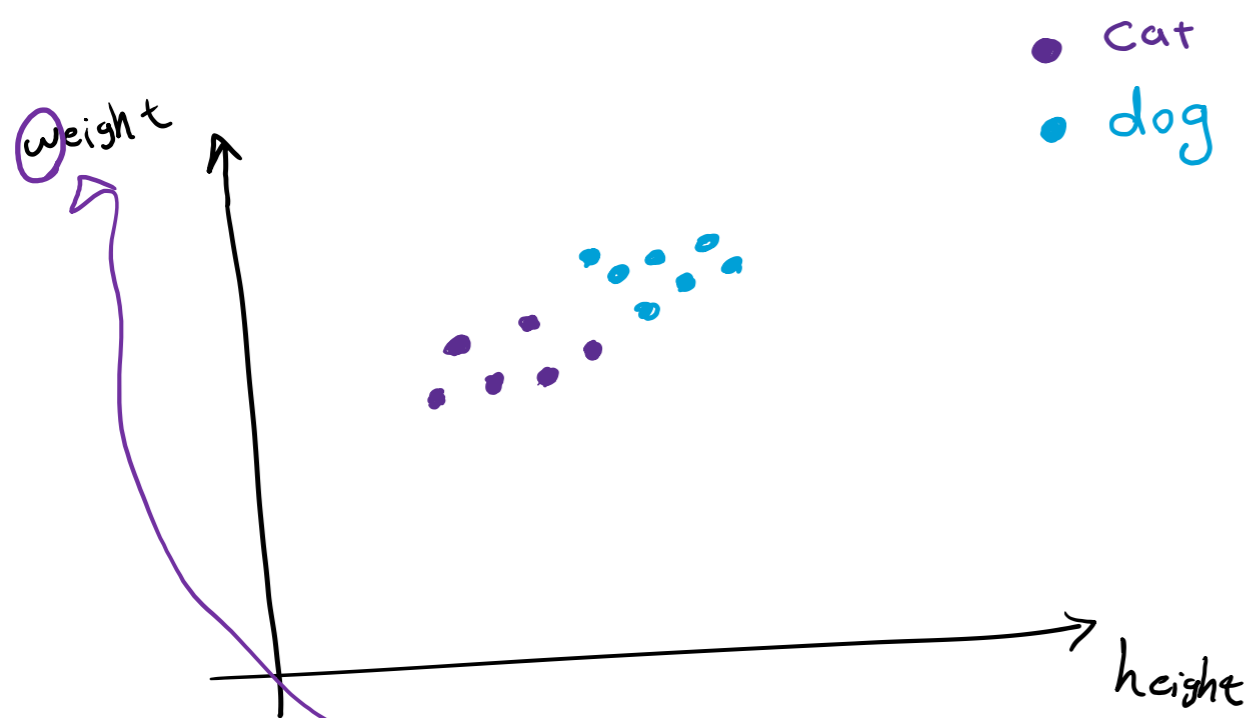
*Ph.D*



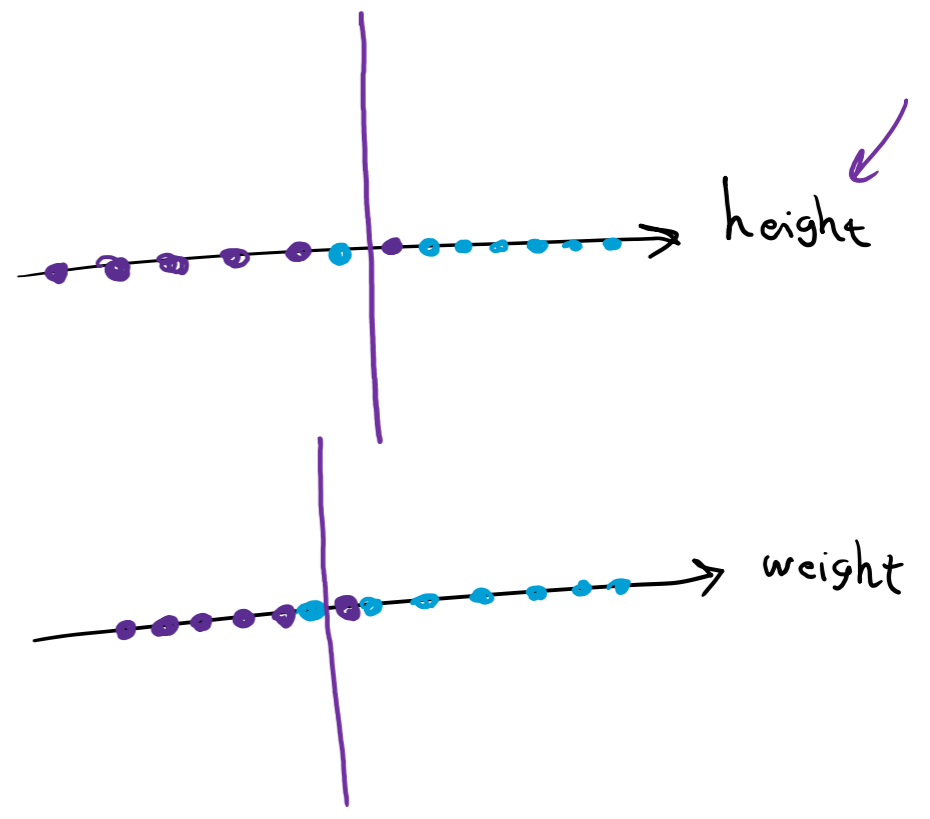
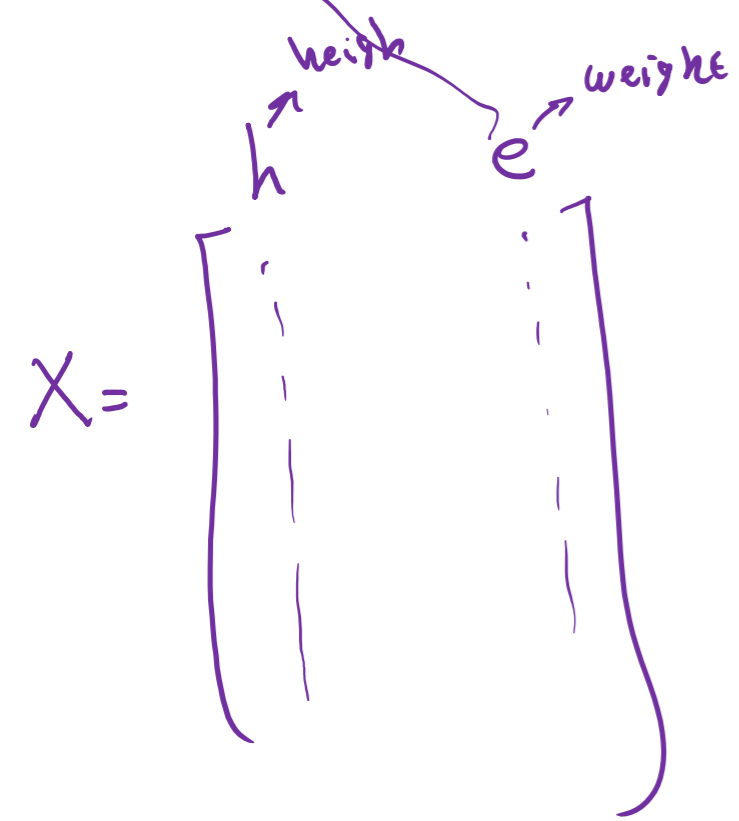
Reality

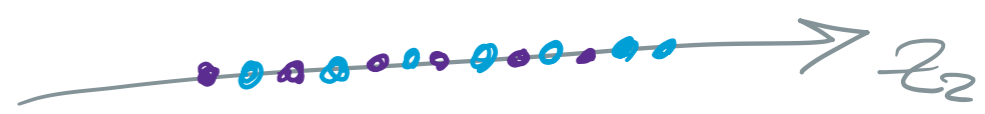
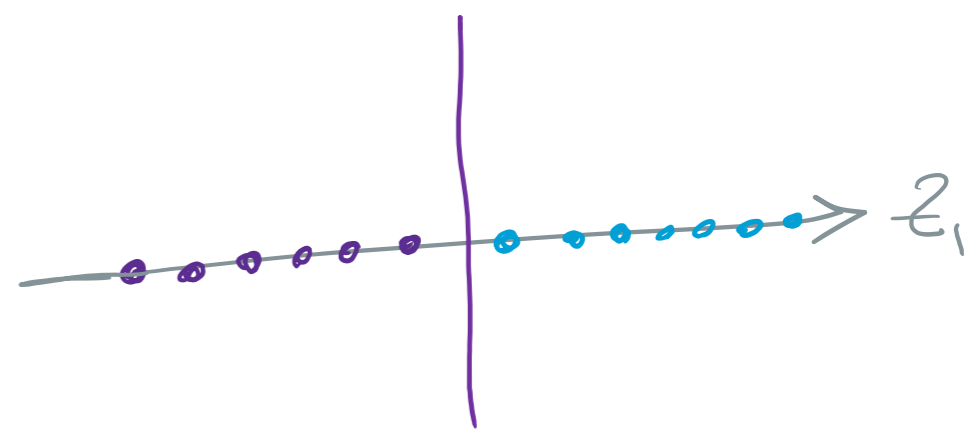
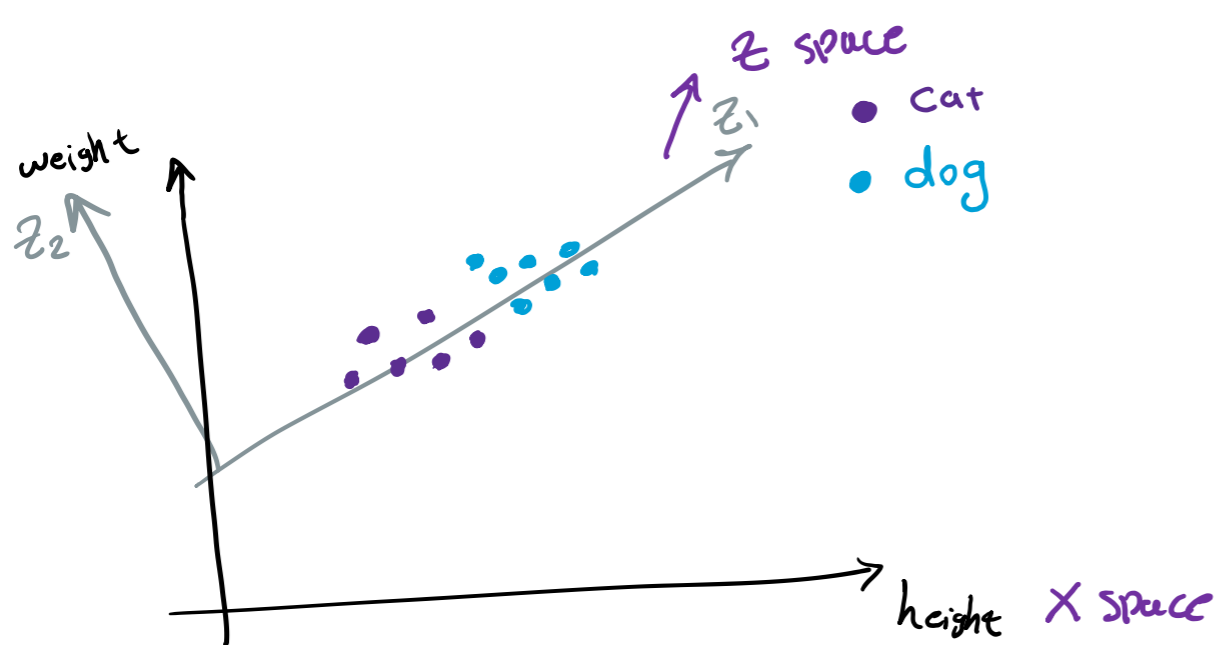
*Ph.D*



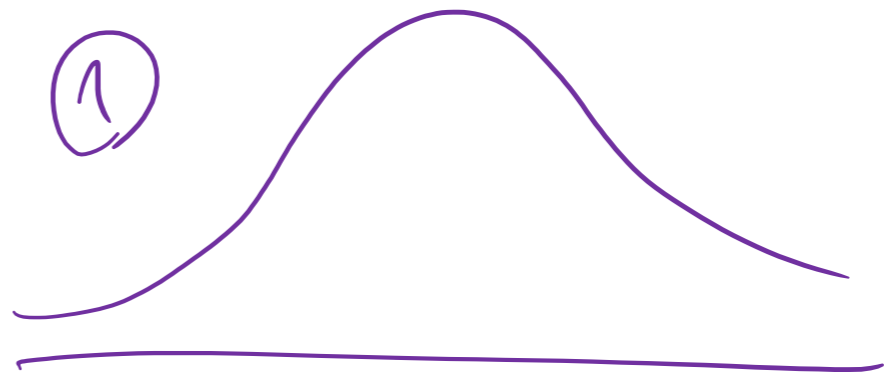
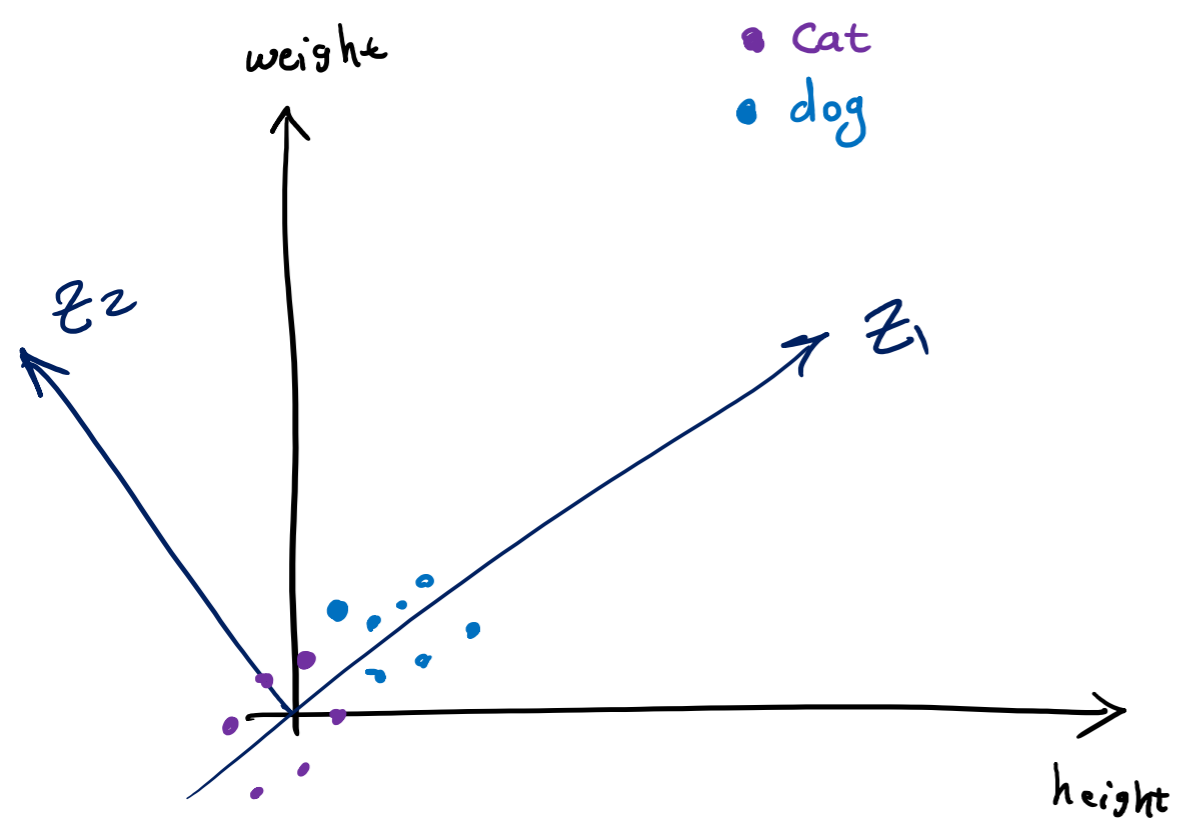


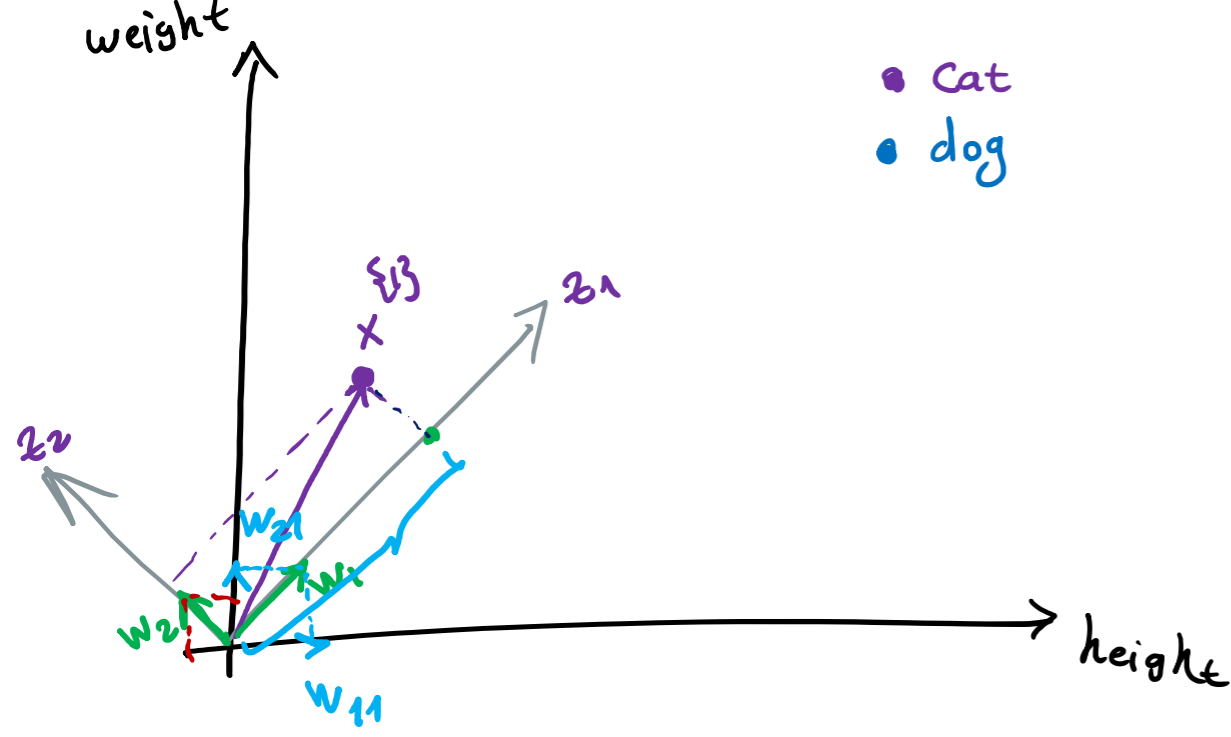
- cat
- dog



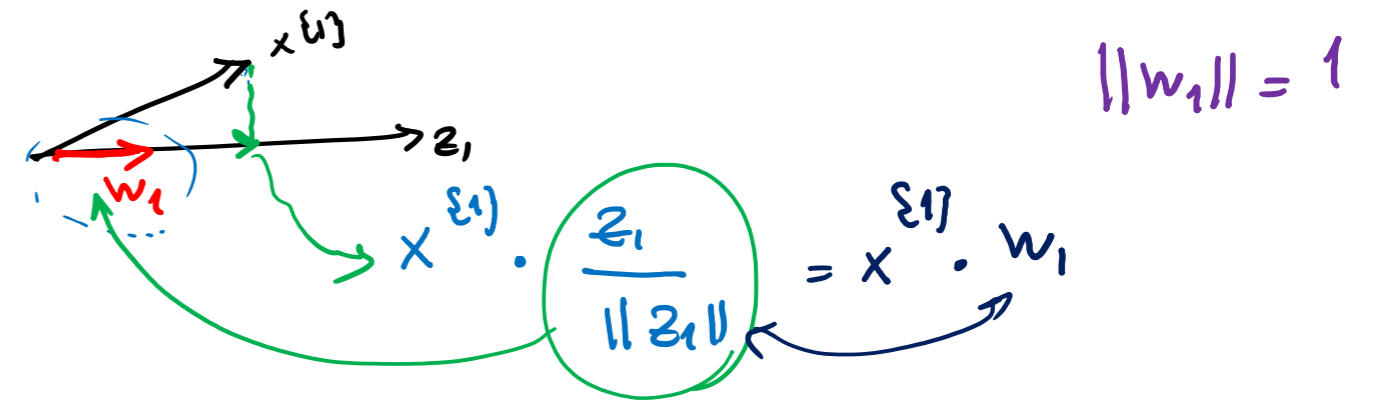


$$X = \begin{bmatrix} h & e \end{bmatrix} \rightarrow \bar{X} = \begin{bmatrix} h - \mu_h & e - \mu_e \end{bmatrix}$$





● Cat  
● dog



$$\|w_1\| = 1$$

$$W = \begin{bmatrix} w_1 & w_2 \\ w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

$$x \xrightarrow{\xi_{13}} z$$

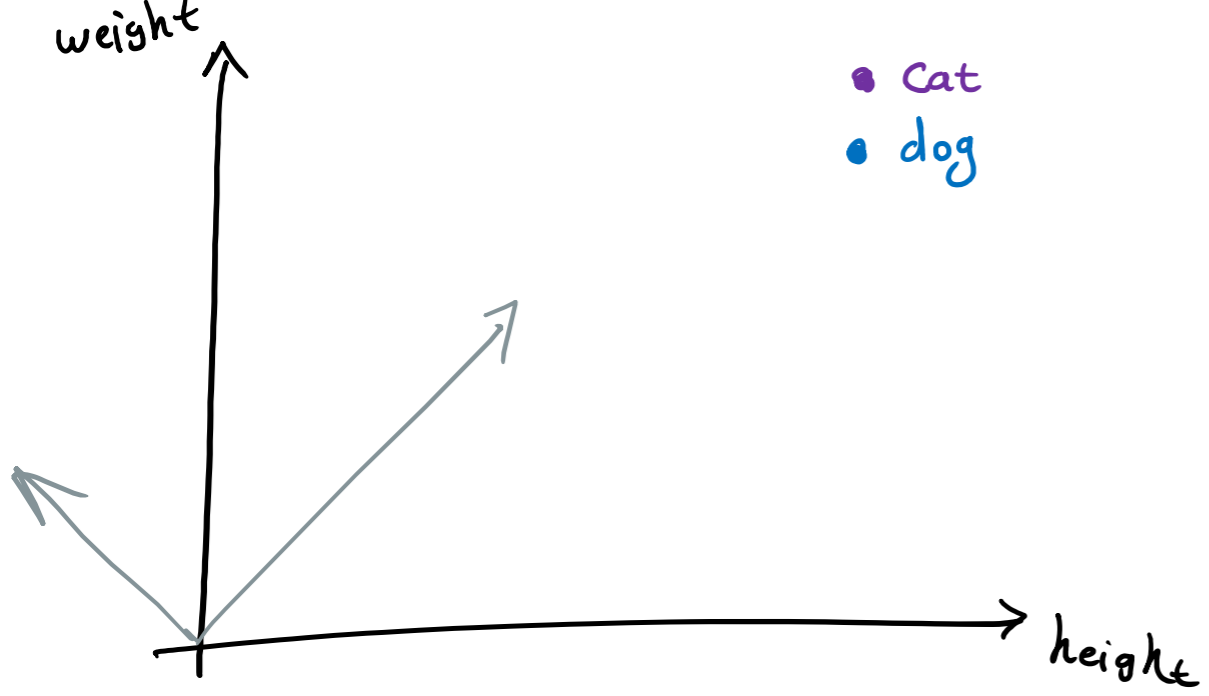
$$x^{\xi_{13}} = [h^{\xi_{13}}, e^{\xi_{13}}]$$

$$z^{\xi_{13}} = [z_1^{\xi_{13}}, z_2^{\xi_{13}}]$$

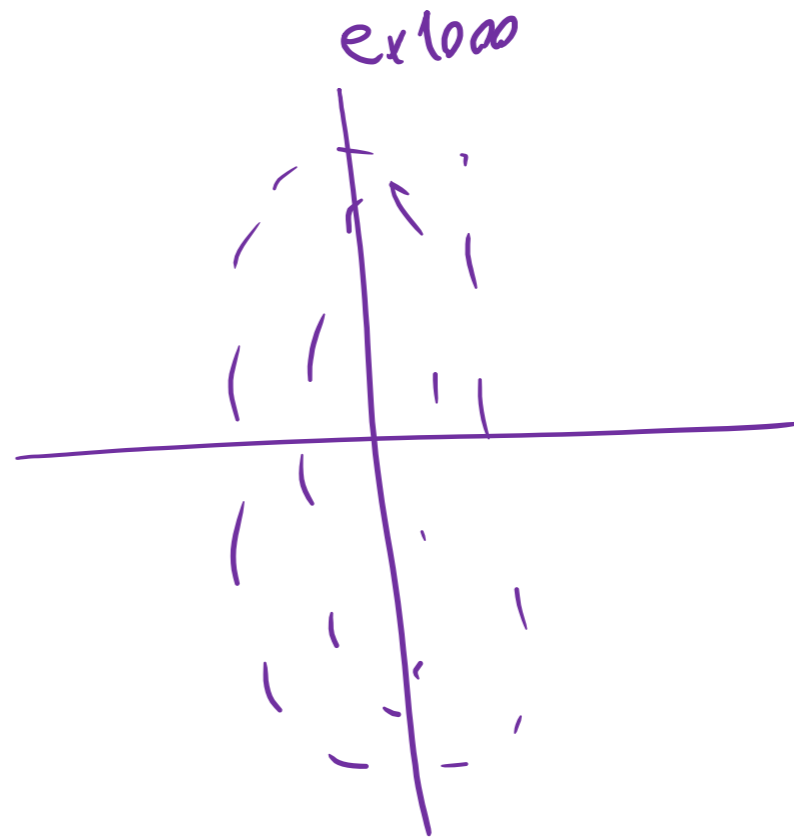
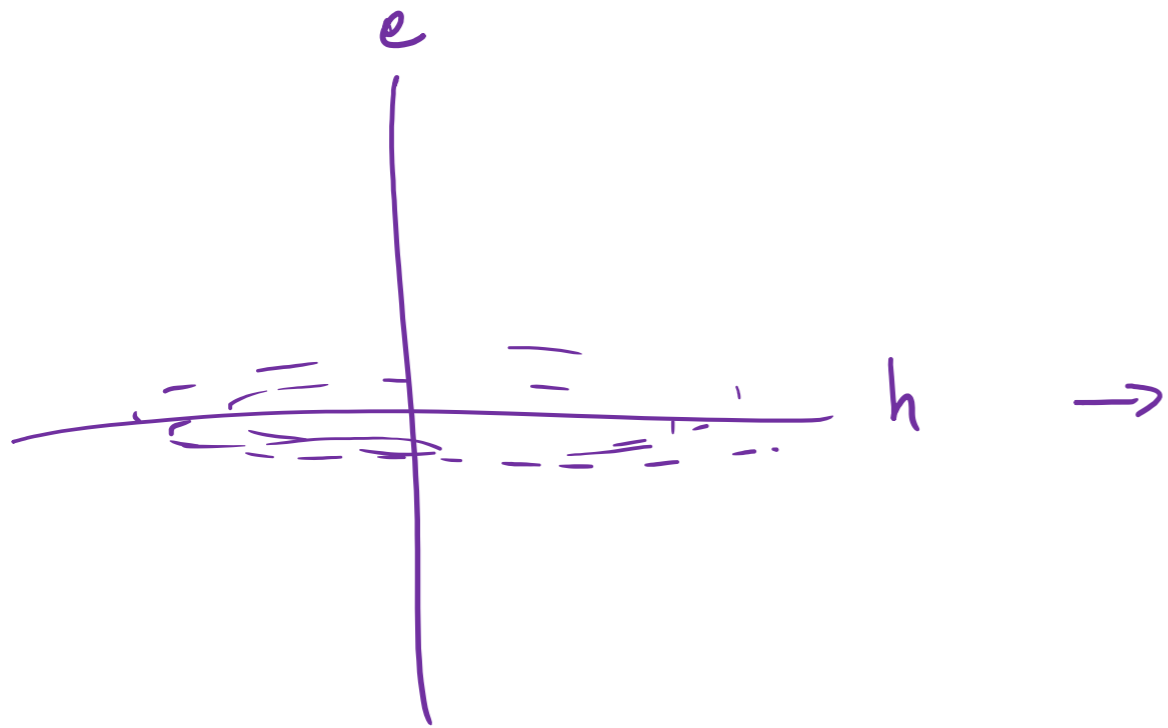
$$z_1^{\xi_{13}} = x^{\xi_{13}} \cdot w_1 = [h^{\xi_{13}}, e^{\xi_{13}}] \begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix} = h^{\xi_{13}} w_{11} + e^{\xi_{13}} w_{21}$$

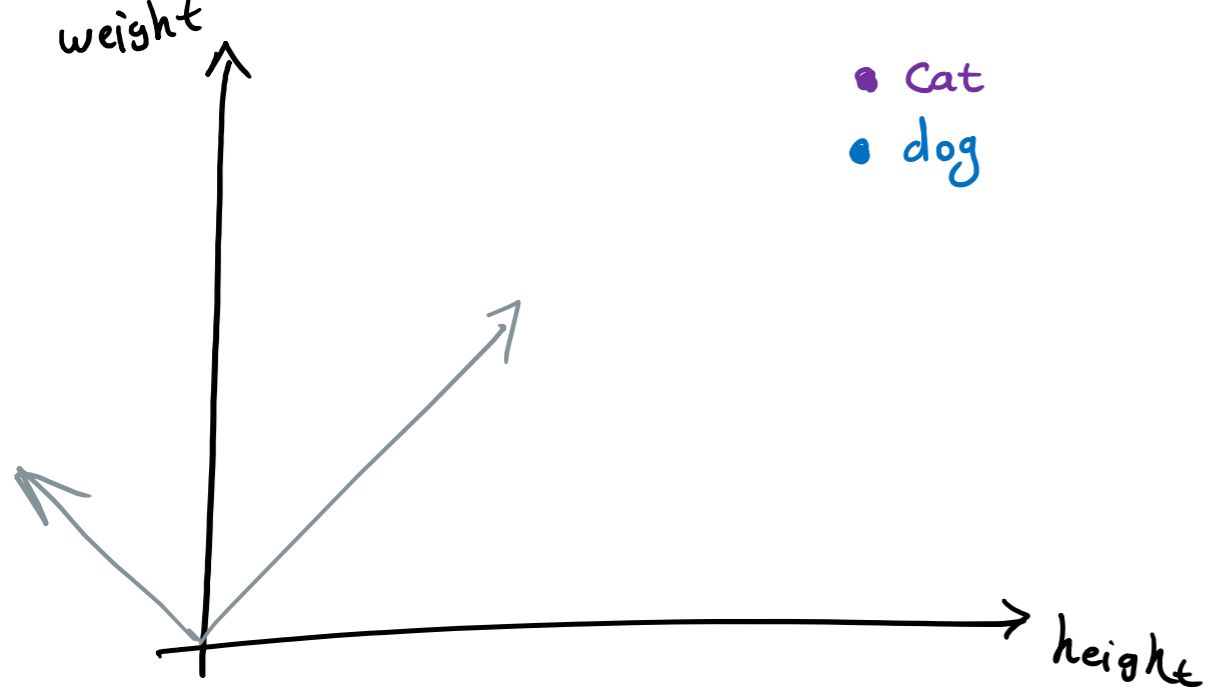
$$z_2^{\xi_{13}} = x^{\xi_{13}} \cdot w_2 = [h^{\xi_{13}}, e^{\xi_{13}}] \begin{bmatrix} w_{12} \\ w_{22} \end{bmatrix} = h^{\xi_{13}} w_{12} + e^{\xi_{13}} w_{22}$$

$$x^{\xi_{13}} = [h^{\xi_{13}}, e^{\xi_{13}}] \rightarrow z^{\xi_{13}} = \begin{bmatrix} h^{\xi_{13}} w_{11} + e^{\xi_{13}} w_{21} & h^{\xi_{13}} w_{12} + e^{\xi_{13}} w_{22} \end{bmatrix}$$



$$\bar{X} \rightarrow \bar{X}^{(*)} = \begin{bmatrix} \frac{h - \mu_h}{\sigma_h} & \frac{e - \mu_e}{\sigma_e} \end{bmatrix}$$





$$X \xrightarrow{\{x_i\}} z$$

$$X \rightarrow X \cdot w \rightarrow z$$

$$w = \begin{bmatrix} w_1 \rightarrow z_1 \\ w_2 \rightarrow z_2 \\ \dots \end{bmatrix}$$

$$\text{Var}(X) = \frac{1}{N} \sum_{i=1}^N (x^{i3} - \mu)^2$$

Maximize  $\text{Var}(z) = \frac{1}{N} \sum_{i=1}^N (x \cdot w - \mu \cdot w)^2$

S.t.  $\|w\|=1$

$$\mu = [\mu_h, \mu_e]$$

$$X = \begin{bmatrix} h & e \\ i & i \end{bmatrix}$$

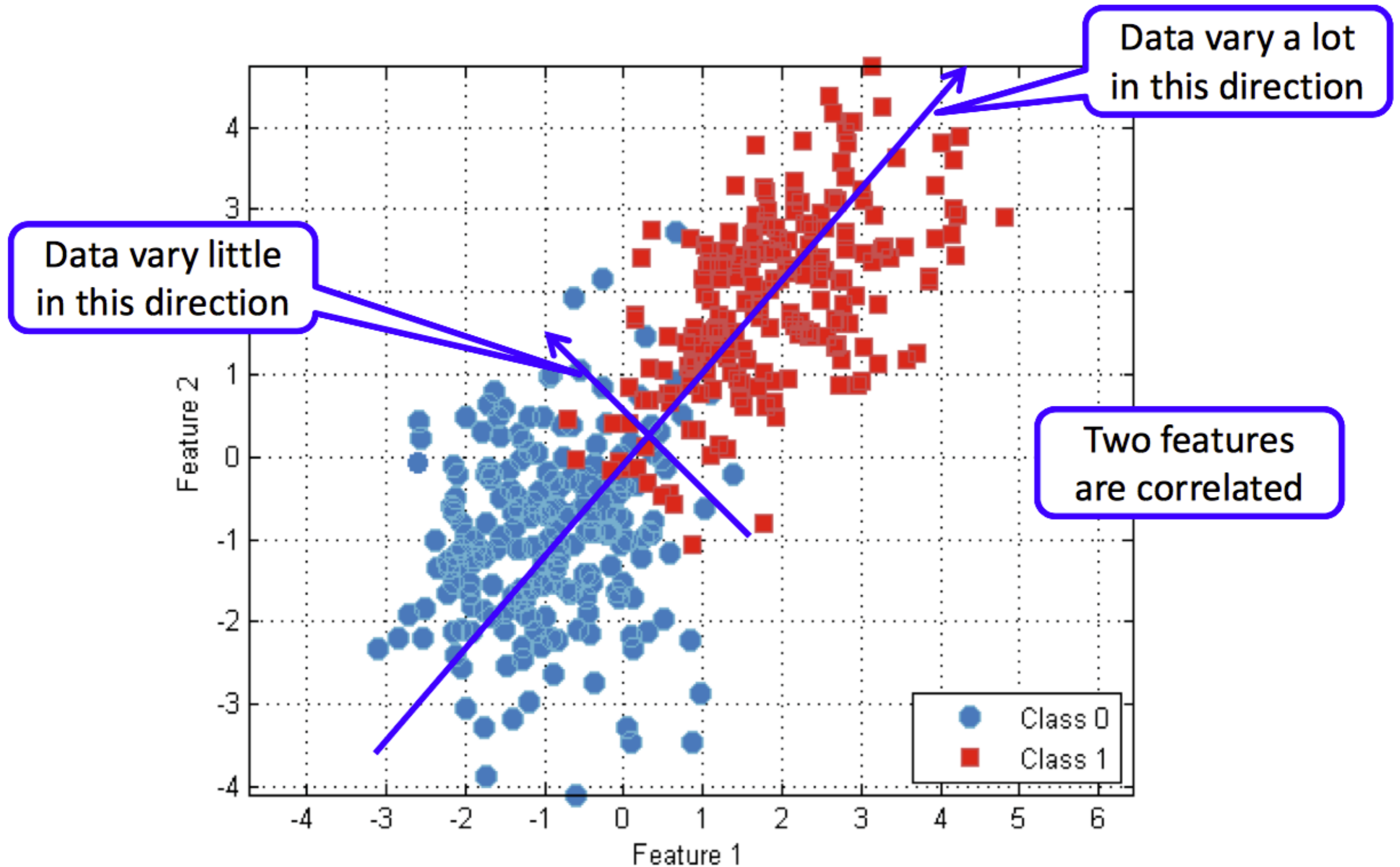
$$w = \begin{bmatrix} w_1 & \dots & w_d \end{bmatrix} \quad \|w\|=1$$



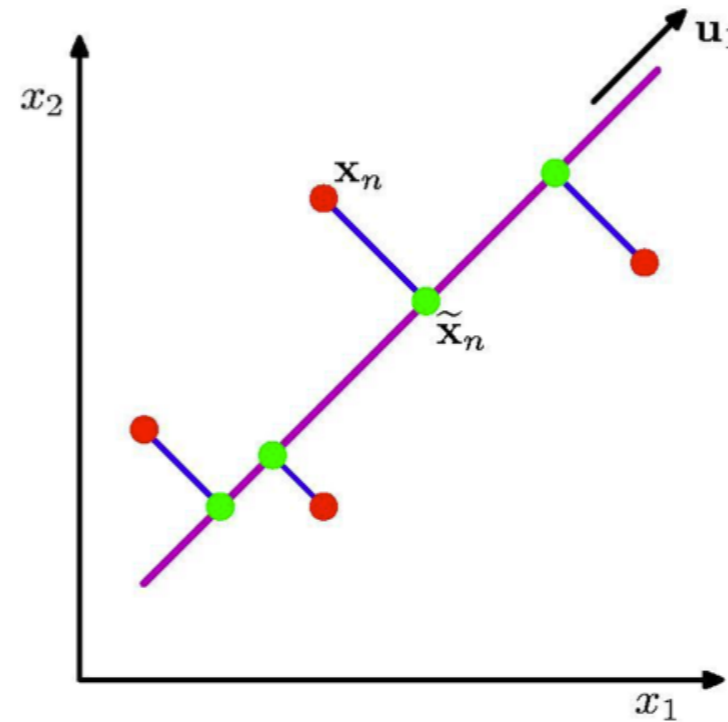
# PCA: Dimension Reduction by Capturing **Variation**

- There are many criteria (geometric based, information theory based, etc.)
- One criterion: want to capture **variation** in data
  - variations are “signals” or information in the data
  - need to normalize each variables first
- In the process, also discover variables or dimensions highly **correlated**
  - represent highly related phenomena
  - combine them to form a stronger signal
  - lead to simpler presentation

# Capturing Variation in Data



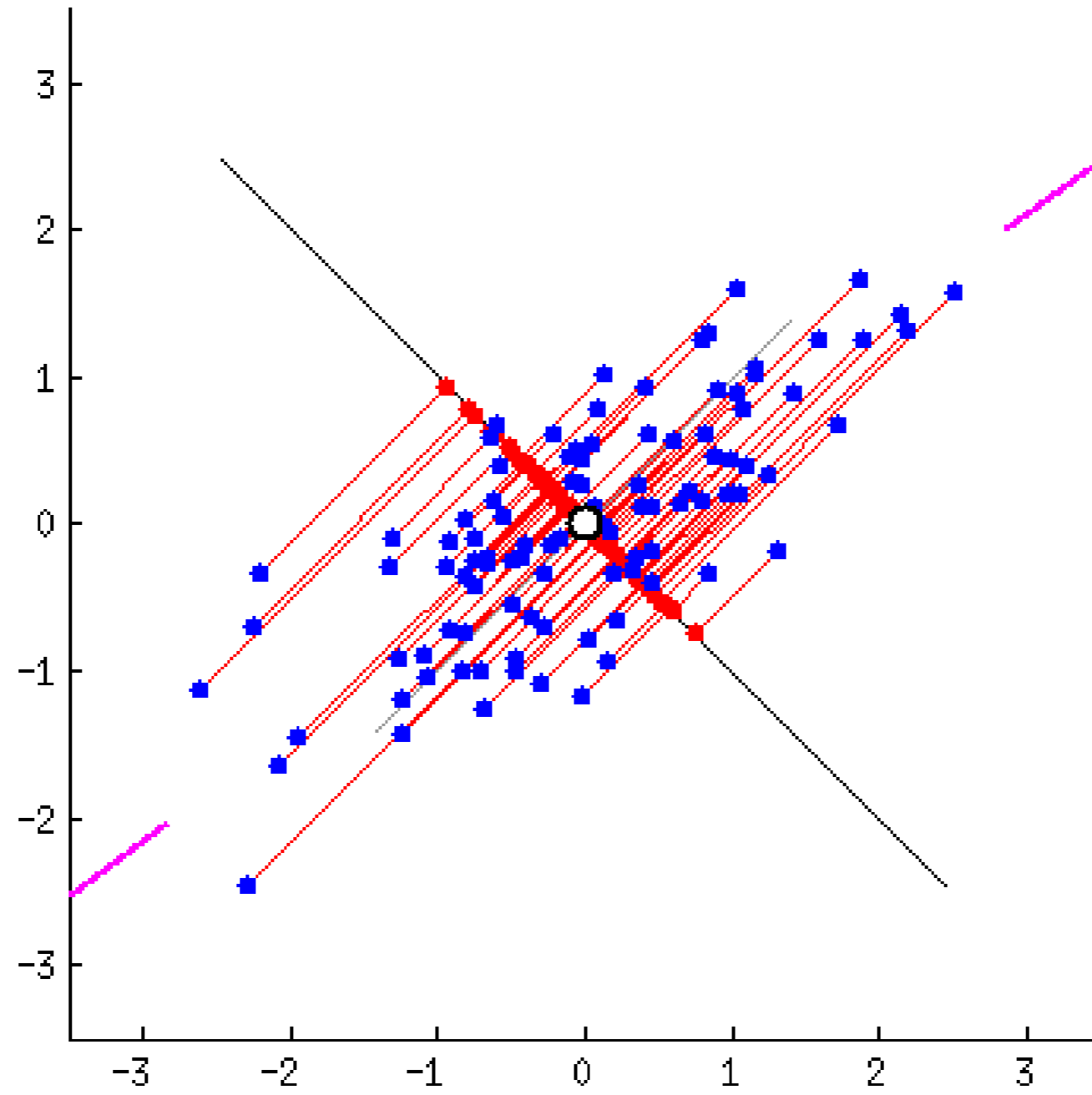
# Two Equivalent Perspectives of PCA



## PCA:

Orthogonal projection of the data onto a lower-dimension linear space that...

- maximizes variance of projected data (purple line)
- minimizes mean squared distance between
  - data point and
  - projections (sum of blue lines)



# Outline

- Overview
- Principle Component Analysis: Main Idea
- The PCA Algorithm ←
- PCA and SVD
- Summary

What is variance equation?

$$\text{Var}(x) = \frac{1}{n} \sum_{i=1}^n (x^{\{i\}} - \mu)^2$$

# Formulating the Problem

- Given  $n$  data point  $\{x^{\{1\}}, x^{\{2\}}, \dots, x^{\{n\}}\} \in R^d$  with their mean  $\mu = \frac{1}{n} \sum_{i=1}^n x^{\{i\}}$
- Find a direction  $w \in R^d$  where

$$\|w\| = \sqrt{\sum_{j \in d} \omega_j^2} = 1$$

We constrain the norm of  $w$  to be equal to one to avoid having very large variance in each new dimension.

- Given  $n$  data point  $\{x^{\{1\}}, x^{\{2\}}, \dots, x^{\{n\}}\} \in R^d$  with their mean  $\mu$

$$\|w\| = \sqrt{\sum_{j \in d} \omega_j^2} = 1 \qquad \mu = \frac{1}{n} \sum_{i=1}^n x^{\{i\}}$$

- Such that the variance (or variation) of the data along direction  $w$  is maximized

$$\max_{\|w\|=1} \frac{1}{n} \sum_{i=1}^n (x^{\{i\}} w - \mu w)^2$$

variance in new feature space

# An Optimization Problem

- Manipulate the objective with linear algebra

$A^2 = AA^T = A^T A$   
 $w = \begin{bmatrix} 0 \\ \vdots \end{bmatrix}_{dx \times d}$

①

$$\text{Var}(z) = \frac{1}{n} \sum_{i=1}^n (x^{i\{i\}} w - \mu w)^2 = \frac{1}{n} \sum_{i=1}^n ((x^{i\{i\}} - \mu)w)^2 =$$

$$= \frac{1}{n} \sum_{i=1}^n \underbrace{((x^{i\{i\}} - \mu)w)^T}_{A} \underbrace{((x^{i\{i\}} - \mu)w)}_B = \frac{1}{n} \sum_{i=1}^n w^T (x^{i\{i\}} - \mu)^T (x^{i\{i\}} - \mu) w$$

$(AB)^T = B^T A^T$   
 $\frac{\sum_{i=1}^n (x^{i\{i\}} - \mu)^T (x^{i\{i\}} - \mu)}{n} = C$

②

$$\text{Var}(z) = w^T C w$$

$$w^T \left( \frac{1}{n} \sum_{i=1}^n (x^{i\{i\}} - \mu)^T (x^{i\{i\}} - \mu) \right) w = w^T C w$$

$(1 \times d)$   $(d \times d)$   $(d \times 1)$   
 $(1 \times 1)$

Covariance matrix

# Let's optimize it

$$\text{Var}(z) = w^T C w$$

$$\text{s.t. } \underbrace{w^T w = 1}_{\|w\|=1} \rightsquigarrow g(w) = w^T w - 1$$

$$L(w, \lambda) = \underbrace{w^T C w} - \lambda (w^T w - 1)$$

$$\frac{\partial L(w, \lambda)}{\partial w} = \underbrace{2Cw} - 2\lambda w = 0$$

$$Cw = \lambda w$$

$$Ax = \lambda x$$

$$\frac{\partial \left( \frac{a}{w^T} \cdot \frac{b}{w} \right)}{\partial w} = a'b + b'a$$

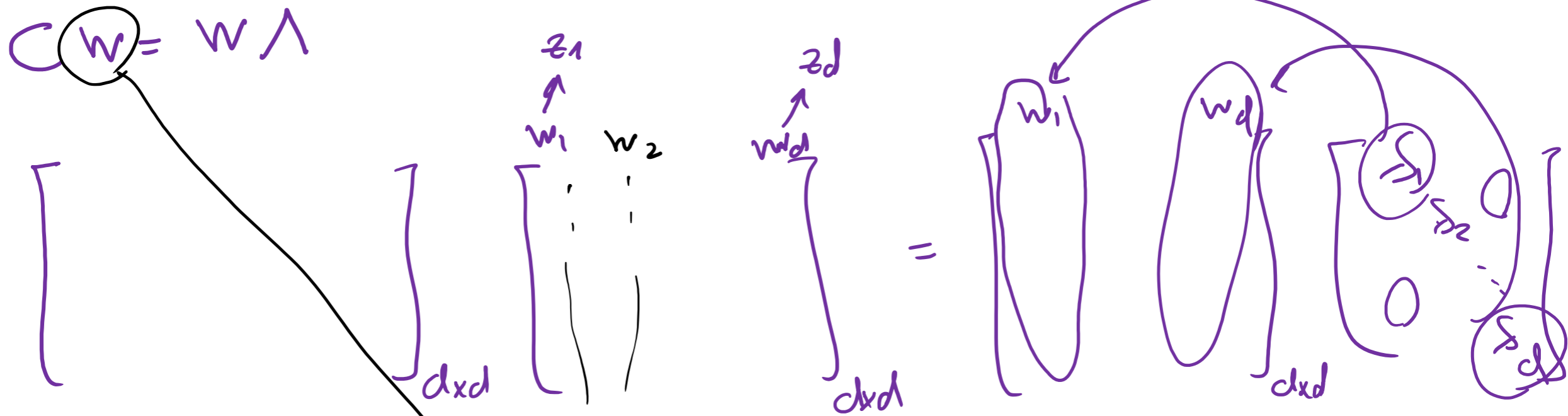
$$Cw \frac{dx}{dx} + \left( \frac{w^T C}{1 \times d} \right)^T \frac{dx}{dx}$$

$$\left[ \begin{matrix} w_1 \\ \vdots \end{matrix} \right]$$

$$Cw + C^T w = w \underbrace{(C + C^T)}_{2C}$$

$$C = \left[ \begin{matrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{matrix} \right] \text{ Symmetrical}$$

$$C = C^T$$



$$w_1 \cdot w_2 = 0$$

$$w_1 \perp w_2$$

$$\text{Var}(z) = W^T C W$$

$$\text{Var}(z) = W^T W \Sigma$$

$$\text{Var}(z) = \Sigma$$

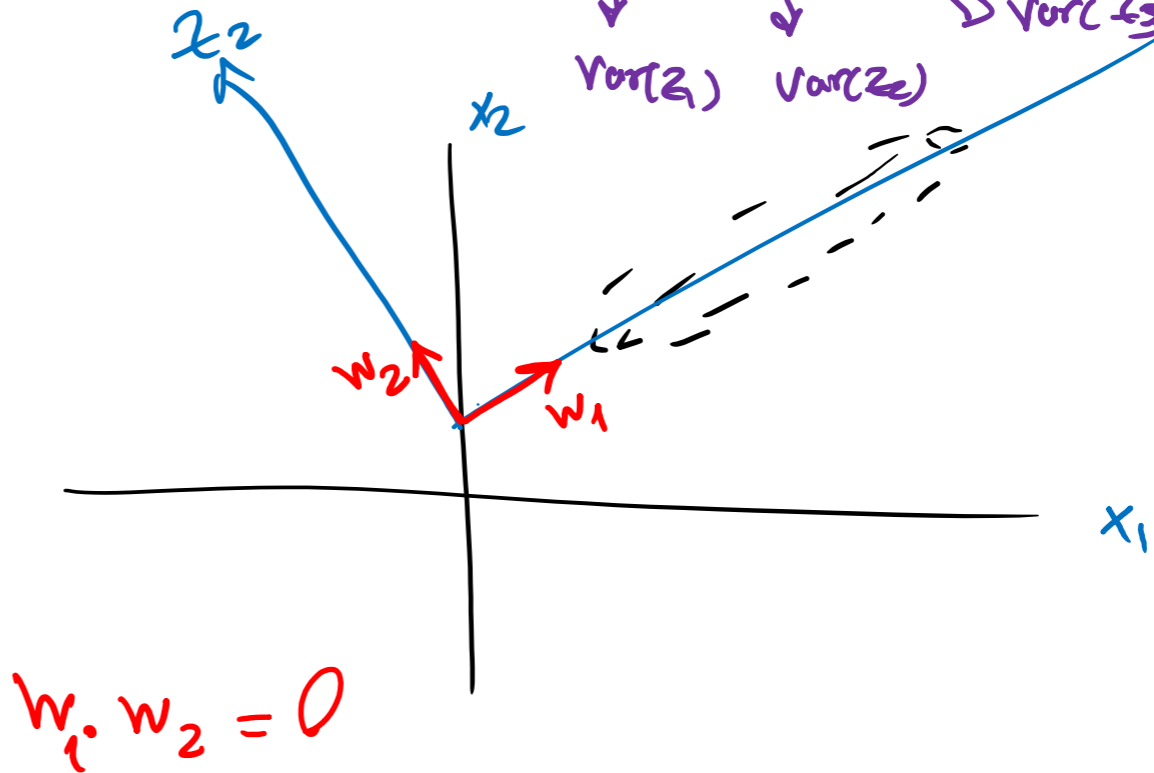
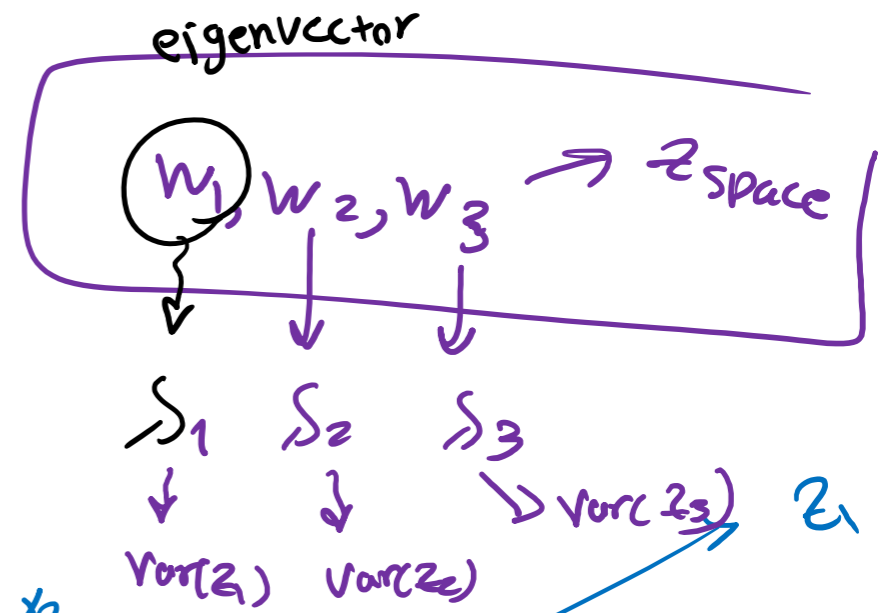
$$C W = \Sigma W = W \Sigma$$

$$X \rightarrow z$$

$$X_i \rightarrow X_i \cdot w_i \rightarrow z_i$$

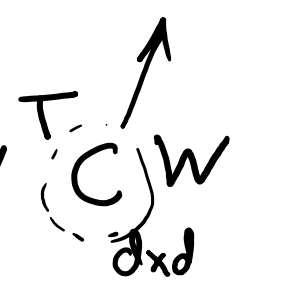
We want preserve 90% Variance  
 hyperparameter

$$\frac{\lambda_1 + \lambda_2 + \lambda_3 + \dots}{\lambda_1 + \lambda_2 + \dots + \lambda_d} \geq 90\%$$



$X \rightarrow Z$       PCA: Maximizing variance  
 $n \times d$        $n \times d$

$$\text{Var}(x) = \frac{1}{N} \sum_{i=1}^N (x^{(i)} - \mu)^2 \Rightarrow \text{Max Var}(z) = \frac{1}{N} \sum_{i=1}^N (x^{(i)} \cdot w - \mu \cdot w)^2 = w^T C w$$

Covariance  


s.t.  $w^T w = 1$

$$C(w) = \sum w \quad C = \frac{\overline{X X^T}}{N}$$

$$\text{Var}(z) = \lambda$$

$$w^T C w = \text{Var}(z)$$

$$\underbrace{w^T w}_{"1"} \sum$$

$$\text{Var}(z) = \lambda$$

# Equivalence to The Eigenvalue Problem

**Objective function:**  $\max_{\|w\|=1} w^T C w$

- Form lagrangian function of the optimization problem

$$L(w, \lambda) = w^T C w + \lambda(1 - w^T w)$$

If  $w$  is a maximum of the original optimization problem, then there exist a  $\lambda$ , where  $(w, \lambda)$  is a **stationary point** of  $L(w, \lambda)$

Therefore:

$$\frac{\partial L(w, \lambda)}{\partial w} = 0 = 2Cw - 2\lambda w \Rightarrow Cw = \lambda w$$

# Eigen-Value Problem

- Eigen-value problem

$d$ : dimension

- Given a symmetric matrix  $C \in R^{d \times d}$

$C$  is also a positive semidefinite matrix

- Find a vector  $w \in R^d$  and  $\|w\| = 1$

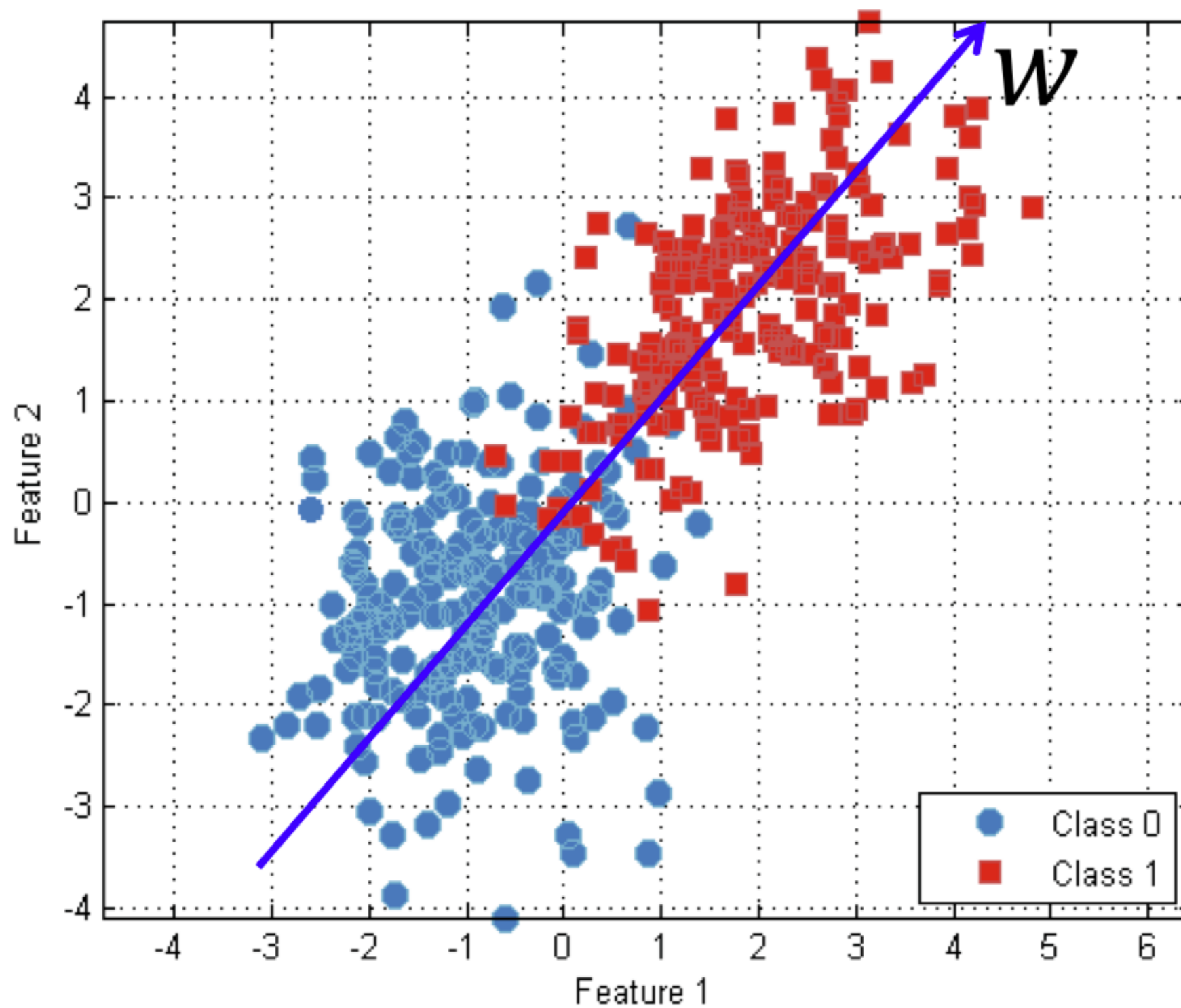
- Such that

$$Cw = \lambda w$$

- There will be multiple solution of  $w_1, w_2, \dots, w_d$  for its corresponding  $\lambda_1, \lambda_2, \dots, \lambda_d$

- They are ortho-normal:  $w_i^T w_i = 1$        $w_i^T w_j = 0$

# Principal Direction of the Data



# Variance in the Principal Direction

- Principal direction  $w$  satisfies

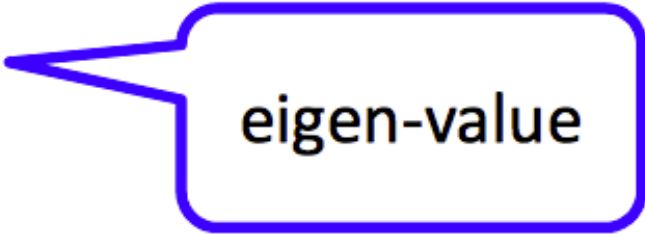
$$Cw = \lambda w = w\lambda$$

- Variance in principal direction is

$$w^T Cw$$

$$= w^T w\lambda$$

$$= \lambda$$

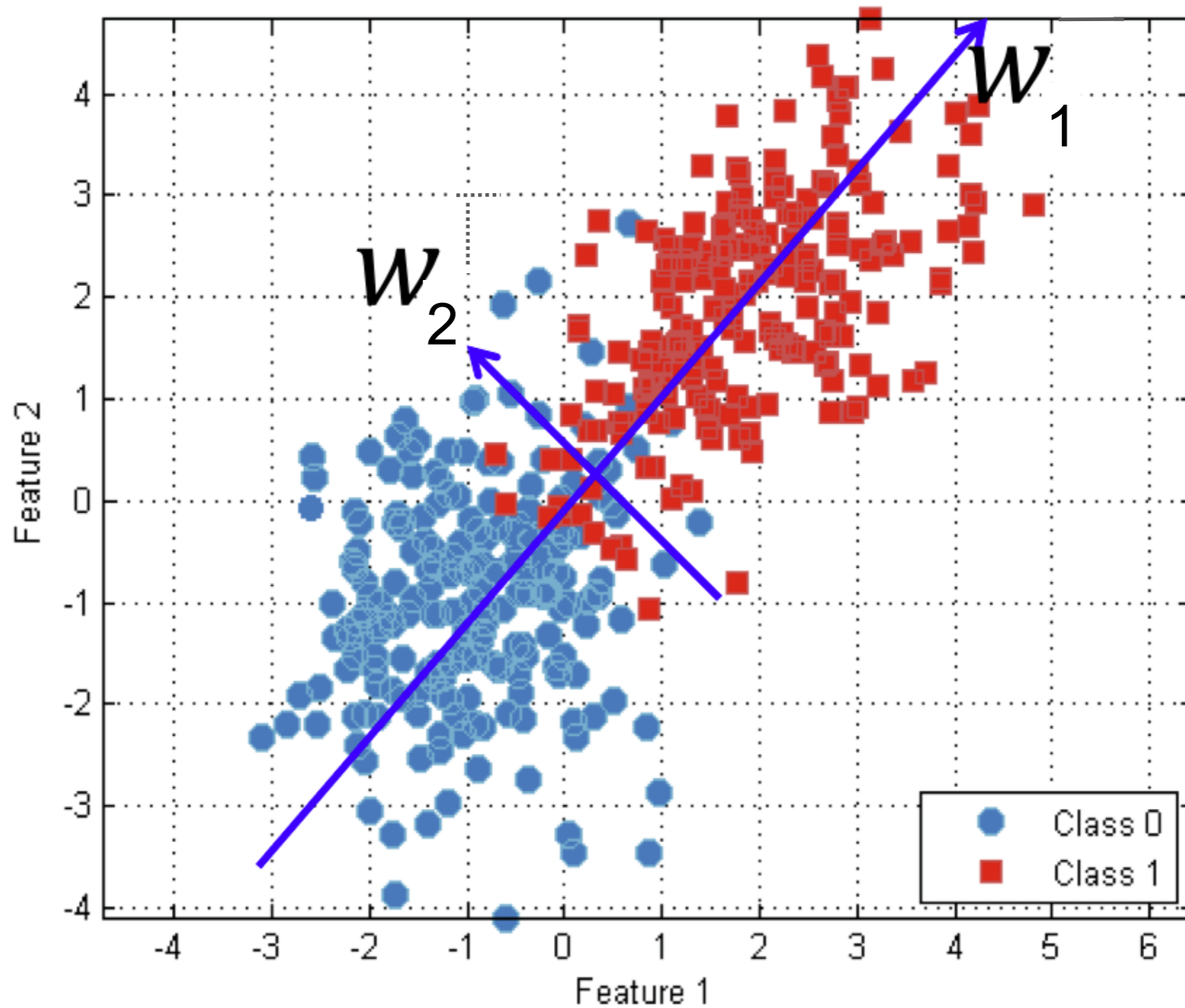


eigen-value

# Multiple Principal Directions

- Directions  $w_1, w_2, \dots$  which has
  - the largest variances
  - but are **orthogonal** to each other
- Take the eigenvectors  $w_1, w_2, \dots$  of  $C$  corresponding to
  - the largest eigenvalue  $\lambda_1$ ,
  - the second largest eigenvalue  $\lambda_2$
  - ...

# Extra Principal Directions



# Relations Between Principal Components

Principal component #1: points in the direction of the **largest variance**.

Each subsequent principal component

- is **orthogonal** to the previous ones, and
- points in the directions of the **largest variance of the residual subspace**

# The PCA Algorithm

- Given  $n$  data points,  $\{x_1, x_2, \dots, x_n\} \in R^d$  with mean

- Step 1: Estimate the mean and covariance matrix from data

$$\mu = \frac{1}{n} \sum_{i=1}^n x^{\{i\}} \quad \text{and} \quad C = \frac{1}{n} \sum_{i=1}^n (x^{\{i\}} - \mu)^T (x^{\{i\}} - \mu)$$

Principal directions


- Step 2: Take the eigenvectors  $w_1, w_2, \dots$  of  $C$  corresponding to the largest eigenvalue  $\lambda_1$ , the second largest eigenvalue  $\lambda_2 \dots$

- Step 3: Compute reduced representation

$$z_i = \left( \frac{(x^{\{i\}} - \mu_1)}{\sigma_1} w_1 \quad \frac{(x^{\{i\}} - \mu_2)}{\sigma_2} w_2 \quad \dots \right) \quad \begin{array}{l} z \Rightarrow n \times k \\ k \ll d \end{array}$$

Normalizing by  
standard deviation

# Outline

- Overview
- Principle Component Analysis: Main Idea
- The PCA Algorithm
- PCA and SVD 
- Summary

# Singular Value Decomposition

$$C = \frac{\bar{X}^T \bar{X}}{N}$$

$$\bar{X}_{n \times d}$$

n: instances

d: dimensions

X is a centered matrix

$$\bar{X} = U \Sigma V^T$$

$U_{n \times n}$  → unitary matrix →  $U \times U^T = I$

$$U^T = U^{-1}$$

$\Sigma_{n \times d}$  → diagonal matrix

$V_{d \times d}$  → unitary matrix →  $V \times V^T = I$

$$X = \begin{bmatrix} u_{1 \times 1} & \dots & \dots & \dots & u_{1 \times n} \\ \vdots & \ddots & \dots & \dots & \vdots \\ \vdots & \vdots & \ddots & \dots & \vdots \\ \vdots & \vdots & \dots & \ddots & \vdots \\ u_{1 \times 1} & \dots & \dots & \dots & u_{n \times n} \end{bmatrix} \times \begin{bmatrix} \Sigma_{1 \times 1} & 0 & \dots & 0 \\ 0 & \ddots & \dots & 0 \\ 0 & 0 & \Sigma_{d \times d} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} v_{1 \times 1} & \dots & \dots & \dots & v_{1 \times d} \\ \vdots & \ddots & \dots & \dots & \vdots \\ \vdots & \vdots & \ddots & \dots & \vdots \\ \vdots & \vdots & \dots & \ddots & \vdots \\ v_{d \times 1} & \dots & \dots & \dots & v_{d \times d} \end{bmatrix}$$

$U$   $n \times n$   $\Sigma$   $n \times d$   $V^T$   $d \times d$

$d < n$   $1 \times 1$   $1 \times d$

$\delta = \frac{\sum u_i^2}{n}$

Principle direction

Matrix compression: K dimensions out of d

According to PCA  $\rightarrow Cw = \lambda w = w\lambda$

$$CV = V \frac{\Sigma^2}{n} (V^T V)$$

$$CV = V \frac{\Sigma^2}{n}$$

np. svd(x) = PCA

$$CW = W\lambda$$

Centering X

Covariance  $C_{d \times d} = \frac{1}{n} \sum_{i=1}^n (x^{(i)} - \mu)^T (x^{(i)} - \mu) = \frac{\bar{X}^T \bar{X}}{n}$

$$(U \Sigma V^T)^T = V \Sigma^T U^T$$

$$\bar{X} = U \Sigma V^T$$

$$C = \frac{\bar{X}^T \bar{X}}{n}$$

$$C = \frac{V \Sigma^T U^T U \Sigma V^T}{n} = \frac{V \Sigma^2 V^T}{n}$$

$$C = \frac{V \Sigma^2 V^T}{n} = V \frac{\Sigma^2}{n} V^T$$

$$CV = V \frac{\Sigma^2}{n} V^T V = V \frac{\Sigma^2}{n}$$

According to Eigen-decomposition definition  $\rightarrow CV = V\Lambda$

$V$  is the eigen vectors of covariance (Principal directions)

$\lambda_i = \frac{\sigma_i^2}{n} \rightarrow$  The eigenvalues of covariance matrix

Let's project the data ( $X$ ) on principal directions:

$$\bar{X}V = U\Sigma V^T V = U\Sigma$$

$\bar{X}V$  is linear combination of the original data (x-space) features

Projection of one instance ( $x$ ) on the first principal direction using  $k$  dimensions

$$p_1 = [u_{1 \times 1} \Sigma_{1 \times 1}, u_{1 \times 2} \Sigma_{2 \times 2}, \dots, u_{1 \times k} \Sigma_{k \times k}]$$

$$p_2 = [u_{2 \times 1} \Sigma_{1 \times 1}, u_{2 \times 2} \Sigma_{2 \times 2}, \dots, u_{2 \times k} \Sigma_{k \times k}]$$

$$U \Rightarrow n \times k$$

$$\Sigma \Rightarrow k \times k$$

Upper left corner

Eigen values  $\lambda = \frac{\Sigma^2}{n}$

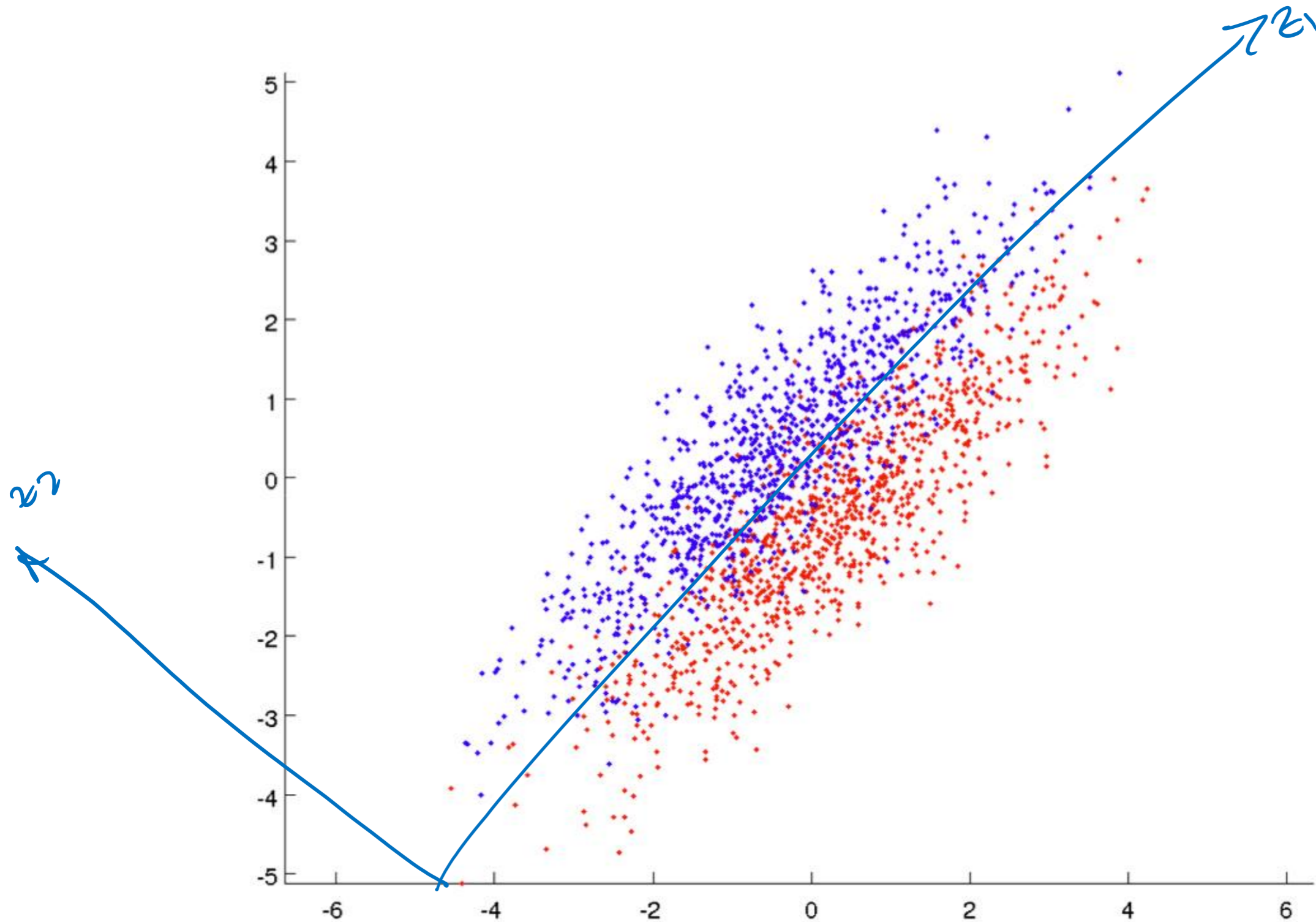
Eigenvectors (principal directions)  $V$

$$\bar{X} = U \Sigma V^T$$

Principal components (Scores) or projections on principal directions

In fact, using the SVD to perform PCA makes much better sense numerically than forming the covariance matrix to begin with, since the formation of  $X^T X$  can cause loss of precision.


# Are Principal Components Good for Classification?



# Why PCA potentially works in classification?

the dimension with the largest variance corresponds to the dimension with the largest entropy and thus encodes the most information (Information Theory). The smallest eigenvectors will often simply represent noise components, whereas the largest eigenvectors often correspond to the principal components that define the data.

# Outline

- Overview
- Principle Component Analysis: Main Idea
- The PCA Algorithm
- PCA and SVD
- Summary 

# Summary

- PCA
  - Finds orthonormal basis for data
  - Sorts dimensions in order of “importance”
  - Discard low significance dimensions
- Uses
  - Get concise low-dimensional representations
  - Remove noise
- Not magic
  - Doesn't know class labels
  - Can only capture linear variations

# Image compression using PCA

PCs # 0



PCs # 10



PCs # 20



PCs # 30



PCs # 40



PCs # 50



→ supervised learning FFS, BFS → Backward Feature selection  
Forward feature selection

$$X = \begin{bmatrix} h & w & a & \dots \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}_{n \times d} \quad Y = \begin{bmatrix} 0 \\ \vdots \\ \vdots \end{bmatrix}_{n \times 1} \quad \text{LR}$$

$$Z = \begin{bmatrix} h \\ \vdots \\ \vdots \end{bmatrix} \quad Y = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \rightarrow \text{LR} \rightarrow \text{acc} = 0.7$$

$$Z = \begin{bmatrix} w \\ \vdots \\ \vdots \end{bmatrix} \quad Y = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \rightarrow \text{LR} \rightarrow \text{acc} = 0.65$$

$$Z = \begin{bmatrix} a \\ \vdots \\ \vdots \end{bmatrix} \quad Y = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix} \rightarrow \text{LR} \rightarrow \text{acc} = 0.4$$

$$Z = \begin{bmatrix} h & w \end{bmatrix} \rightarrow LR \rightarrow \text{acc} = .85$$

$$Z = \begin{bmatrix} h & a \end{bmatrix} \rightarrow LR = .70$$

Backward

$$X = \begin{bmatrix} h & a & w & \dots \end{bmatrix} \quad y = \begin{bmatrix} \end{bmatrix}$$

RF feature selection