The week ahead

- **Quiz 6:** mean is 91% and average completion time 4min 36sec!
- Assignment 2 due Oct 7th 11:59pm (midnight)
- Assignment 3 out Oct 7th
- Quiz 7, Friday, Oct 9th 6am until Oct 10th 11:59am (noon)
  - PCA and linear regression

Important notices

- Office hours sign-up sheet
- Lecture recordings
- Focus videos

CS4641B Machine Learning

Lecture 13: Dimensionality reduction

Rodrigo Borela ▸ rborelav@gatech.edu

These slides are adopted based on slides from Le Song, Chao Zhang, Barnabás Póczos and Mahdi Roozbahani

Outline

- Overview
- Principle component analysis: main idea
- The PCA algorithm
- PCA and SVD
- Summary

- *Complementary reading: Bishop PRML – Chapter 12, Section 12.1*

Outline

- Overview
- Principle component analysis: main idea
- The PCA algorithm
- PCA and SVD
- Summary

Motivation

53 blood and urine samples (features) from 65 people (datapoints)

Instances

	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

Features

Difficult to see the correlations of different features

Motivation

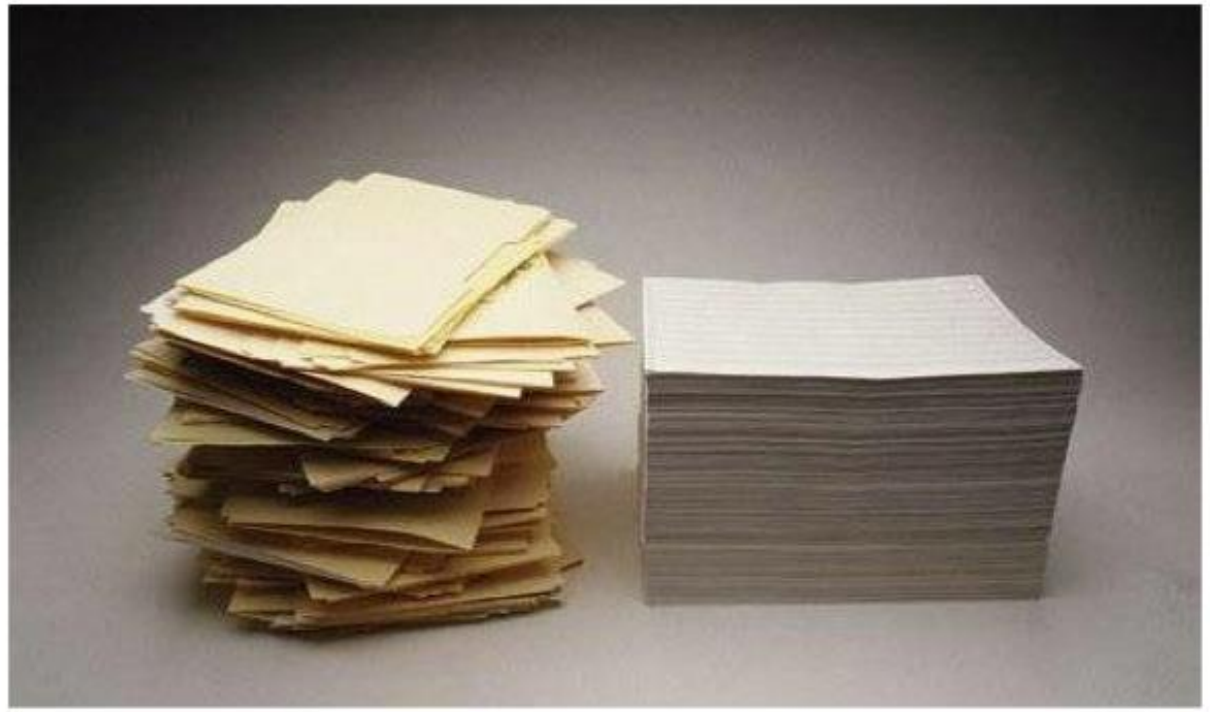
- Is there a better representation than the coordinate axes?

- Is it really necessary to show all the 53 dimensions?
  - What if there are strong correlations between some of the features?

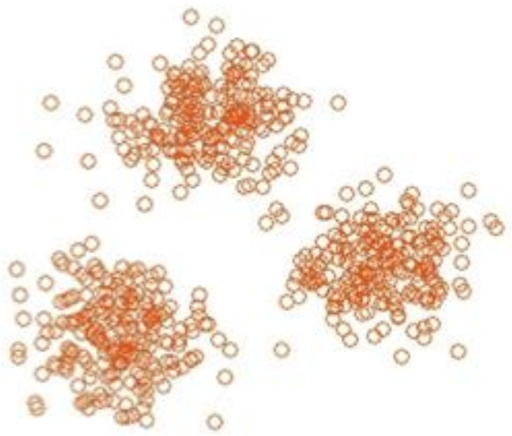
- How could we find the smallest subspace of the 53-D space that keeps the most information about the original data?

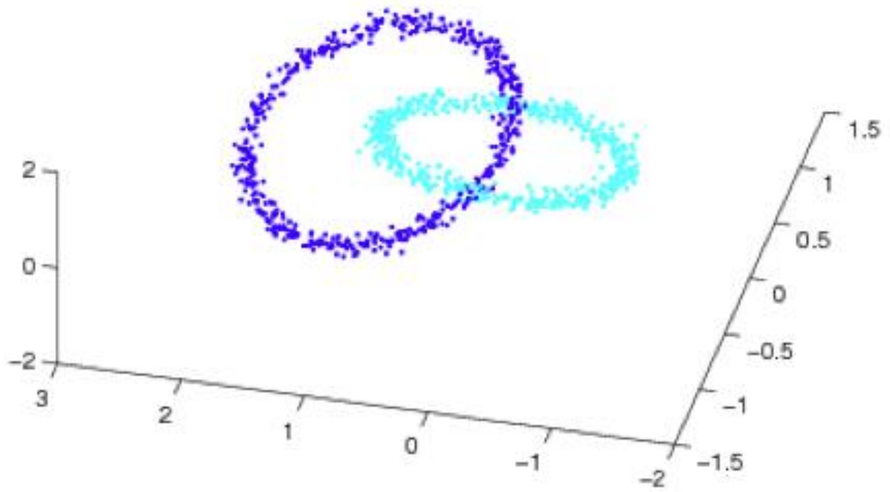
Solution: dimensionality reduction

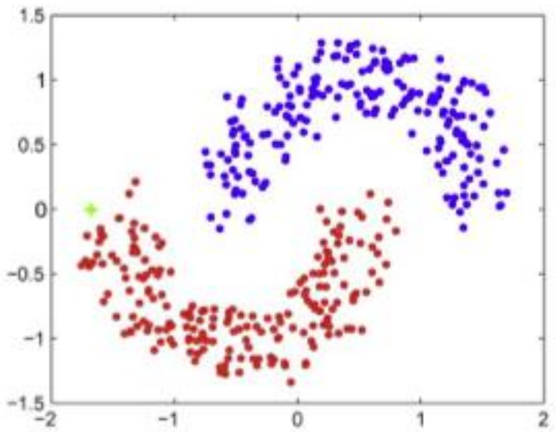
Example: dimensionality reduction for text



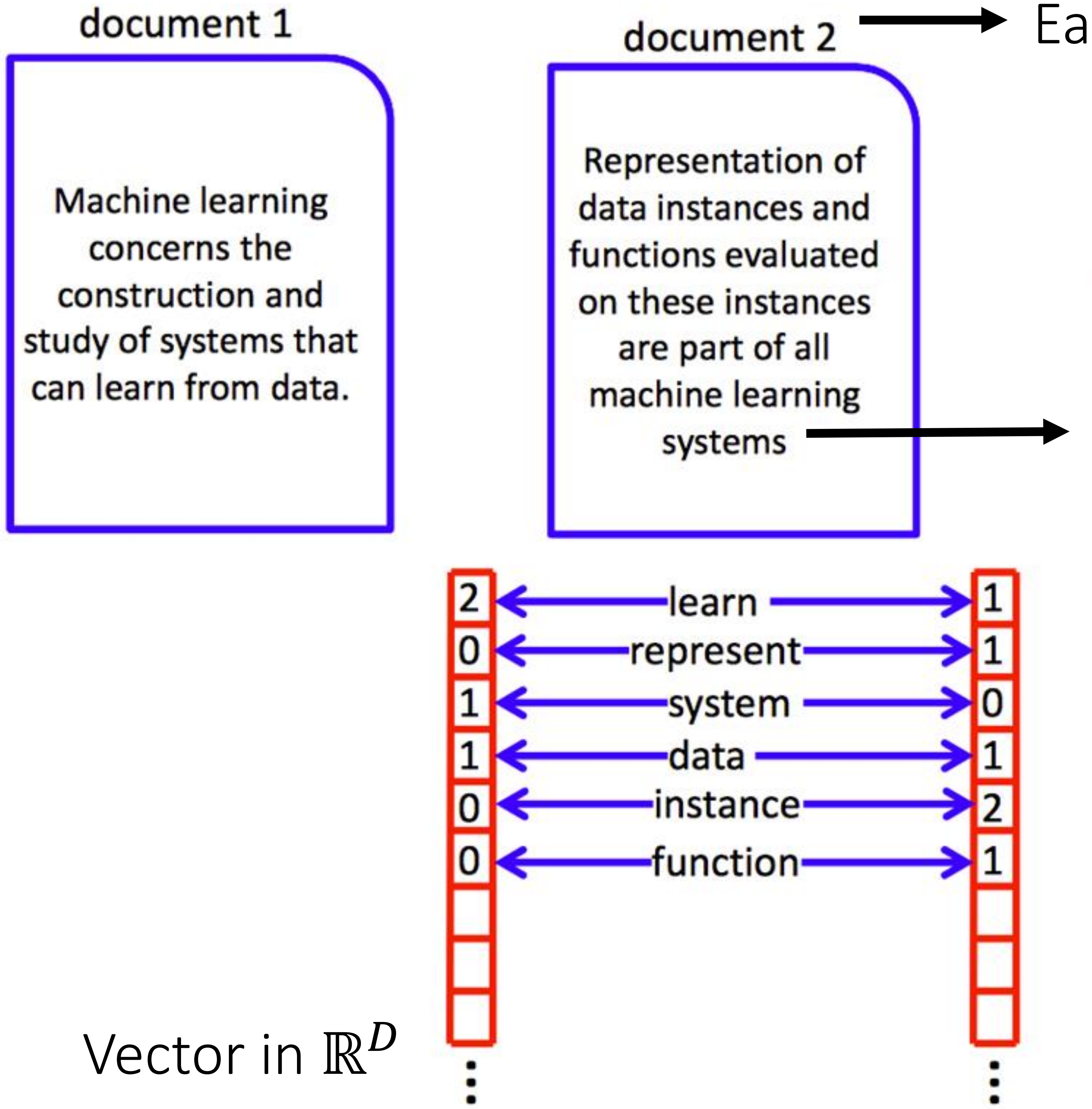
What are the relations between data points?







Bag-of-words representations



Each document is a datapoint

...

Each word is a feature

Vector in $\mathbb{R}^D$

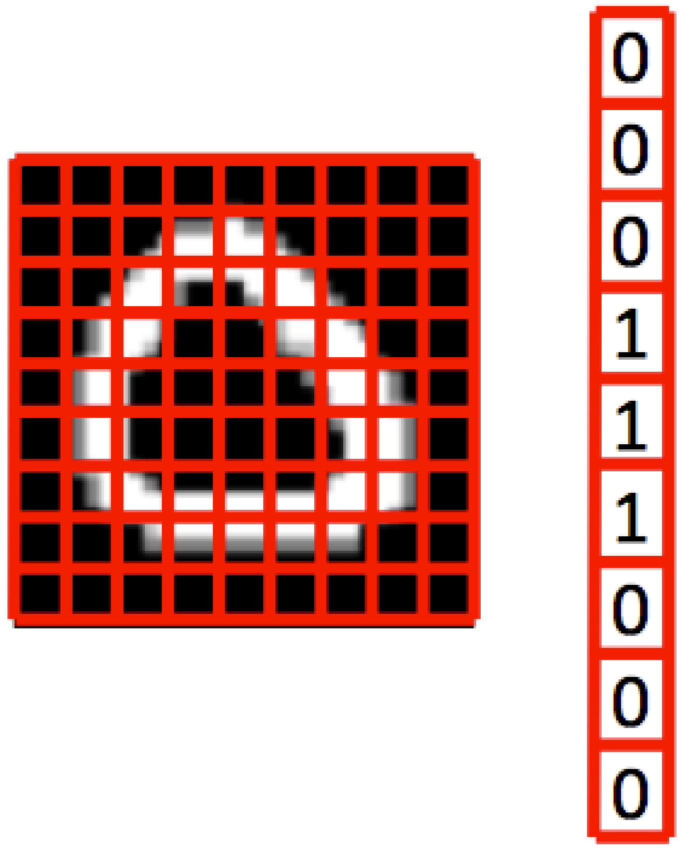
Bag-of-words: term-document data matrix

	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

What is dimensionality reduction?

- The process of reducing the number of random variables under consideration
  - Feature selection, combination or transformation
  - Linear or nonlinear operations



vector in $\mathbb{R}^D$

Original datapoint

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_D \end{pmatrix} \quad \xrightarrow{f(\mathbf{x}): \mathbb{R}^D \to \mathbb{R}^M} \quad \mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \dots \\ z_M \end{pmatrix}$$

Reduced representation

$$M \ll D$$

Applications dimensionality reduction

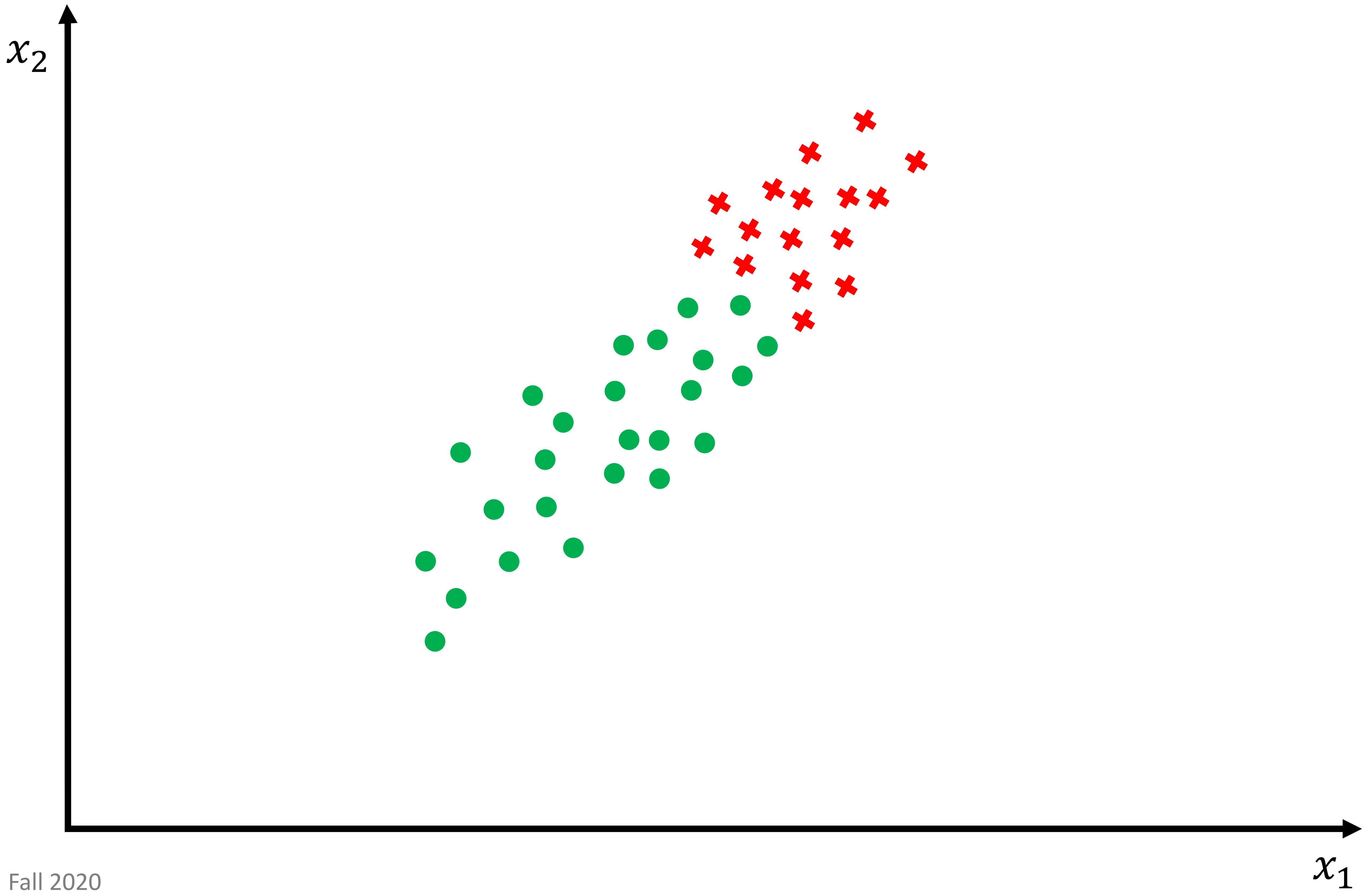
- The dimension-reduced data can be used for:
  - Visualizing, exploring and understanding the data
  - Aggregating weak signals in the data
  - Cleaning the data
  - Speeding up subsequent learning task
  - Building simpler model later

- Key questions of a dimensionality reduction algorithm:
  - What is the criterion for carrying out the reduction process?
  - What are the algorithm steps?

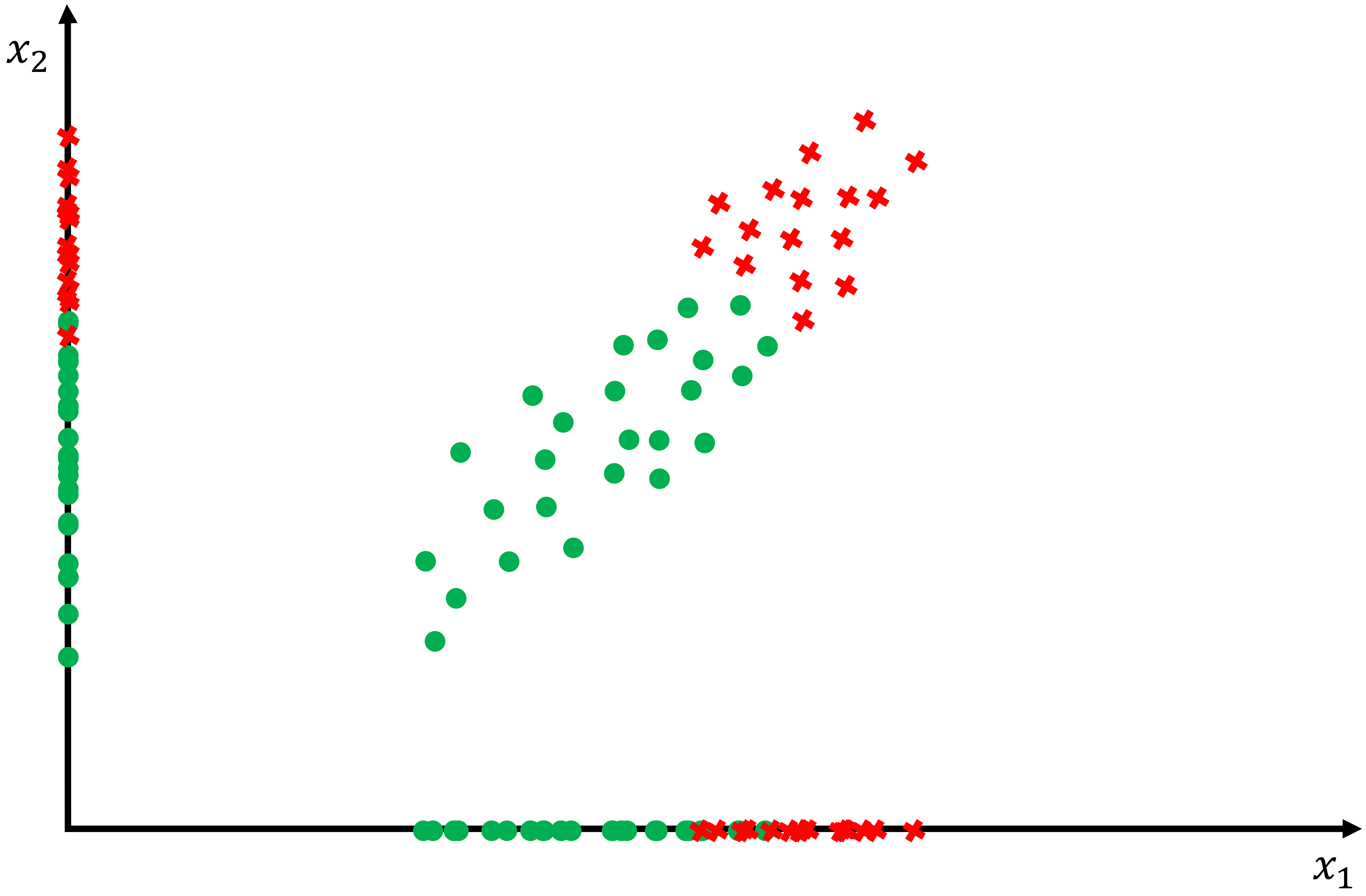
Outline

- Overview
- Principle component analysis: main idea
- The PCA Algorithm
- PCA and SVD
- Summary

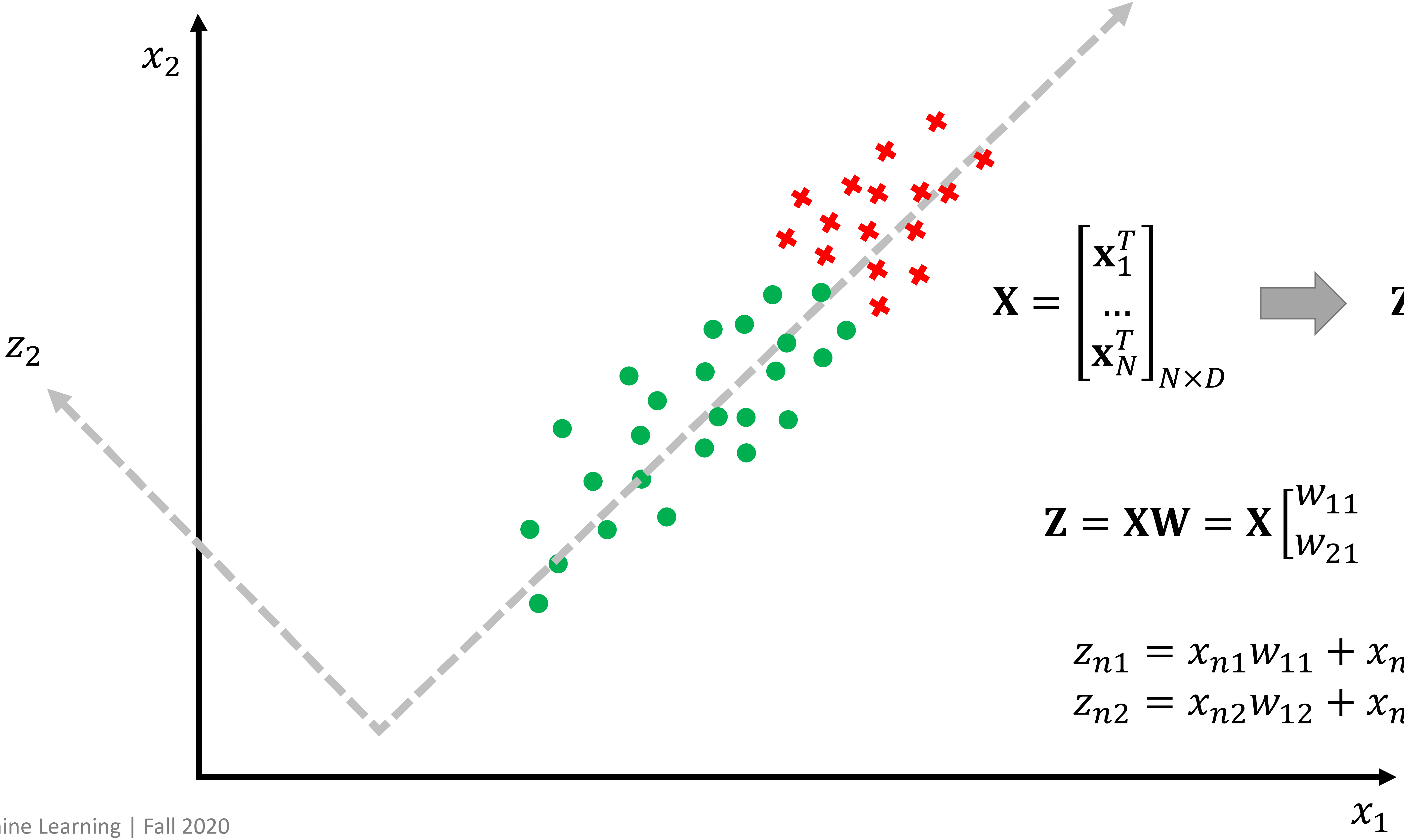
Example: classification



Example: classification



Example: classification

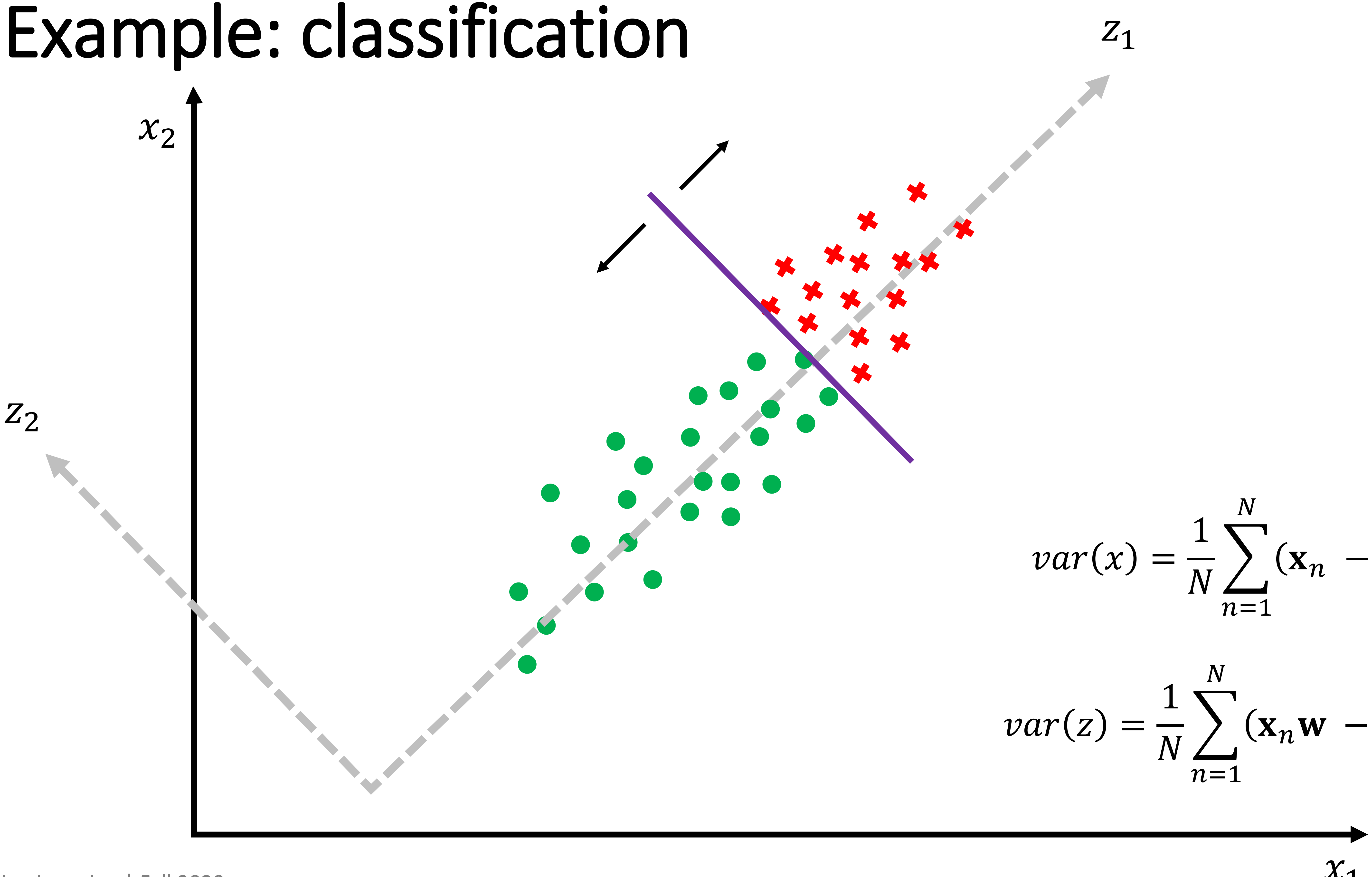


$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \dots \\ \mathbf{x}_N^T \end{bmatrix}_{N \times D} \Rightarrow \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1^T \\ \dots \\ \mathbf{z}_N^T \end{bmatrix}_{N \times D}$$

$$\mathbf{Z} = \mathbf{XW} = \mathbf{X} \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}$$

$$z_{n1} = x_{n1}w_{11} + x_{n2}w_{21}$$
$$z_{n2} = x_{n2}w_{12} + x_{n2}w_{22}$$

Example: classification



$$var(x) = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^2$$

$$var(z) = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n\mathbf{w} - \boldsymbol{\mu}\mathbf{w})^2$$

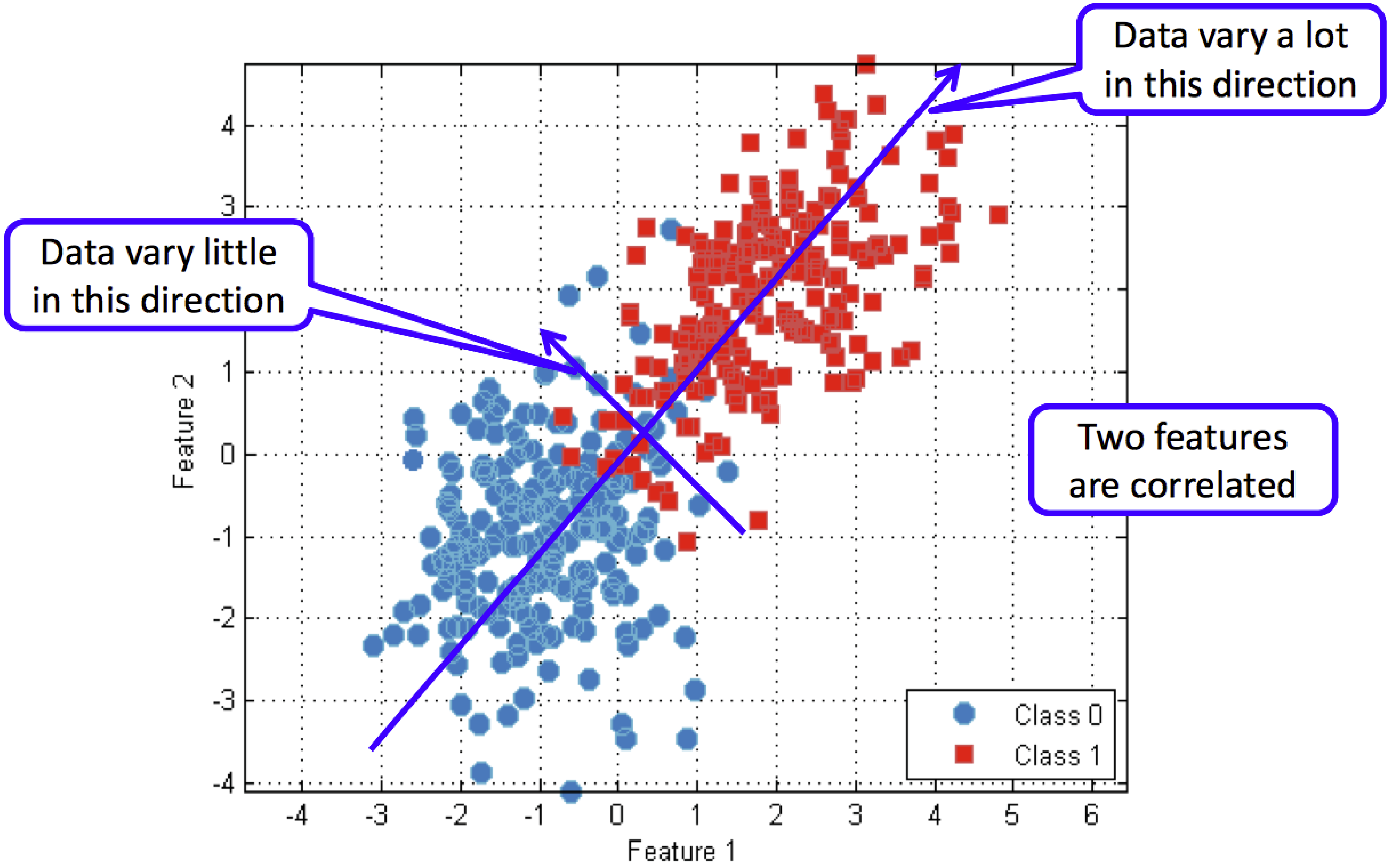
PCA: Dimension reduction by capturing variation

- There are many criteria (geometric based, information theory based, etc.)

- One possible criterion: capture variation in the data
  - Variations are “signals” or information in the data
  - Need to normalize each variable first

- In the process, also discover variables or dimensions that are highly correlated
  - Represent highly related phenomena
  - Combine them to form a stronger signal
  - Lead to simpler presentation

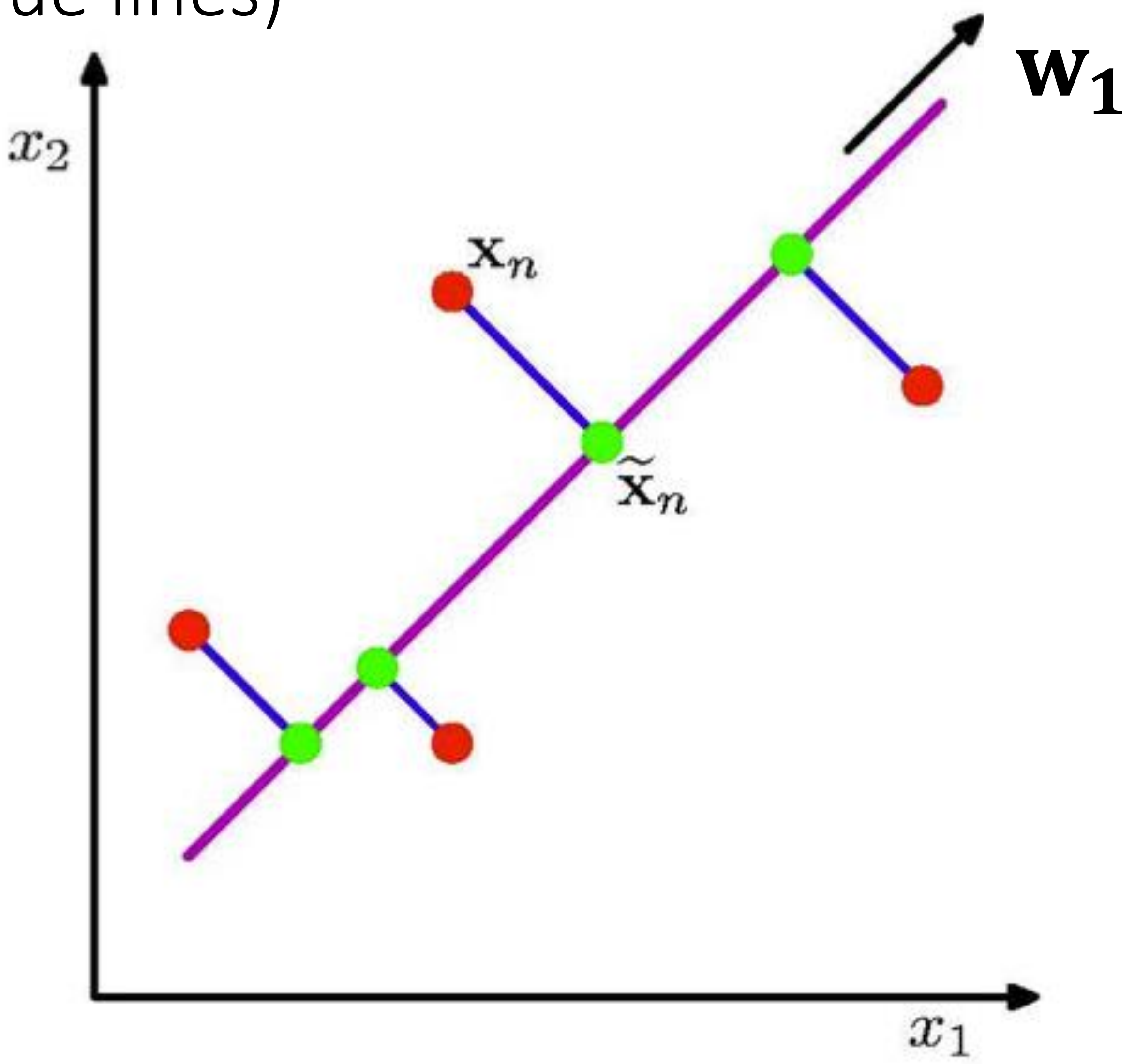
Capturing variation in data



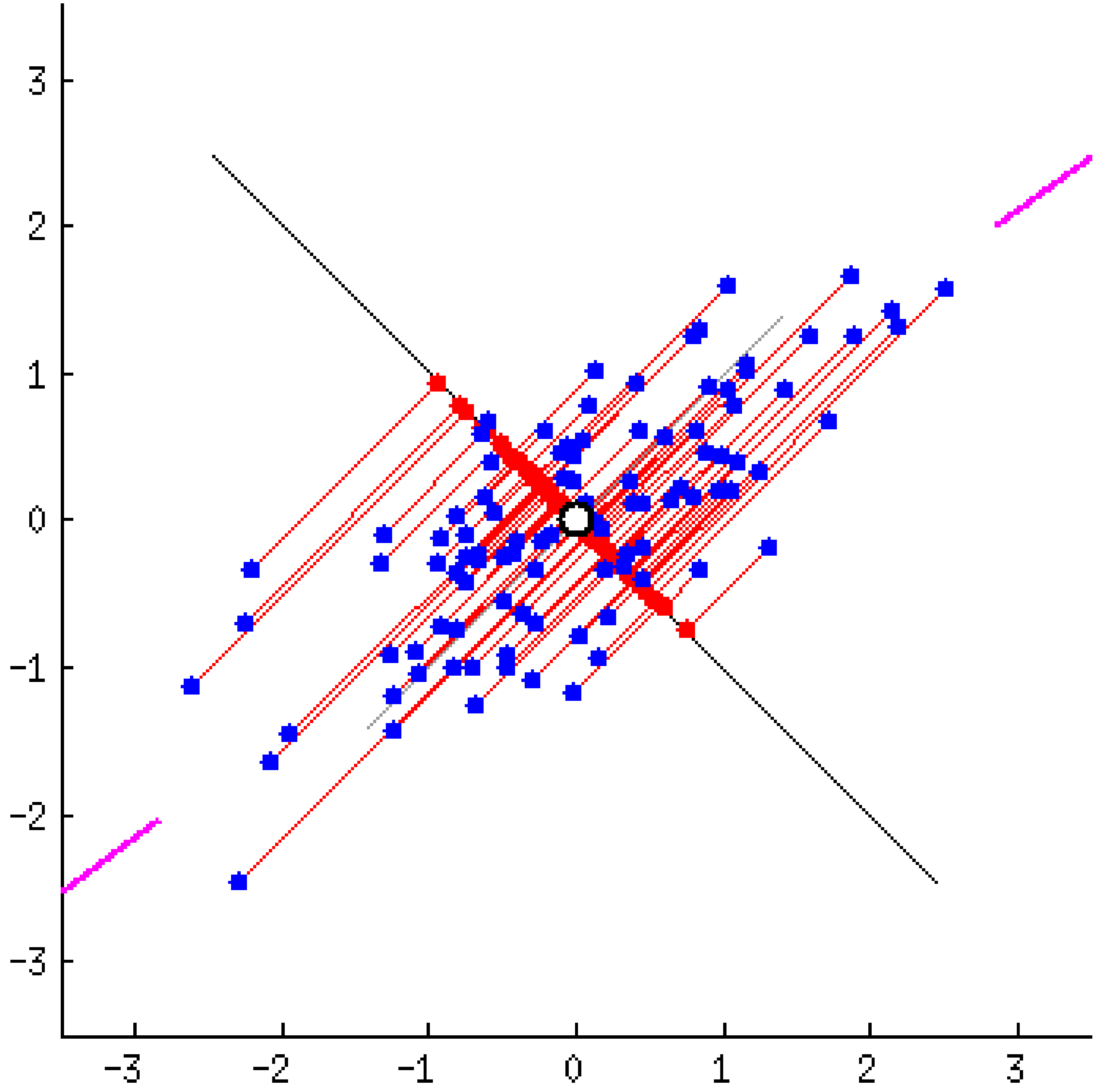
Two equivalent perspectives of PCA

- Orthogonal projection of the data onto a lower-dimension linear space that:
  - Maximizes variance of project data (purple line)

- Minimizes mean squared distance between
  - Data point
  - Projections (sum of blue lines)



Example: iterative algorithm for PCA



Outline

- Overview
- Principle component analysis: main idea
- The PCA algorithm
- PCA and SVD
- Summary

Formulating the problem

- Given N data points, $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathbb{R}^D$ with their mean:

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n}^{N} \mathbf{x}_n$$

- Find direction $\mathbf{w} \in \mathbb{R}^D$ where:

$$\|\mathbf{w}\|_2 = \sqrt{\sum_{d \in D} w_d^2} = 1$$

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

$$\max_{\|\mathbf{w}\|=1} \underbrace{\frac{1}{N}\sum_{n=1}^{N} (\mathbf{x}_n\mathbf{w} - \boldsymbol{\mu}\mathbf{w})^2}_{\text{variance in new feature space}}$$

Formulating the problem

- Manipulate the objective with linear algebra

$$\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n\mathbf{w}-\boldsymbol{\mu}\mathbf{w})^2 \quad = \frac{1}{N}\sum_{n=1}^{N}((\mathbf{x}_n-\boldsymbol{\mu})\mathbf{w})^2 \quad = \frac{1}{N}\sum_{n=1}^{N}\big((\mathbf{x}_n-\boldsymbol{\mu})\mathbf{w}\big)^T((\mathbf{x}_n-\boldsymbol{\mu})\mathbf{w})$$

(remember that $(AB)^T = B^T A^T$)

$$\frac{1}{N}\sum_{n=1}^{N}\mathbf{w}^T(\mathbf{x}_n-\boldsymbol{\mu})^T\,(\mathbf{x}_n-\boldsymbol{\mu})\mathbf{w}$$

$$\mathbf{w}^T\left(\frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n-\boldsymbol{\mu})^T\,(\mathbf{x}_n-\boldsymbol{\mu})\right)\mathbf{w} = \mathbf{w}^T\mathbf{C}\mathbf{w}$$

Equivalence to the eigenvalue problem

- Optimization problem

$$\max_{\|\mathbf{w}\|_2=1} \mathbf{w}^T\mathbf{C}\mathbf{w}$$

- We can rewrite the constraint as follows:

$$\|\mathbf{w}\|_2 = 1 \rightarrow (\|\mathbf{w}\|_2)^2 = 1^2 \rightarrow \mathbf{w}^T\mathbf{w} = 1 \rightarrow 1 - \mathbf{w}^T\mathbf{w} = 0$$

- Form Lagrangian function of the optimization problem

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

- If $\mathbf{w}$ is a maximum of the original optimization problem then there exists a λ where $(\mathbf{w}, \lambda)$ is a stationary point of $L(\mathbf{w}, \lambda)$, therefore:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \rightarrow 2\mathbf{C}\mathbf{w} - 2\lambda\mathbf{w} = 0 \rightarrow \mathbf{C}\mathbf{w} = \lambda\mathbf{w}$$

Equivalence to the eigenvalue problem

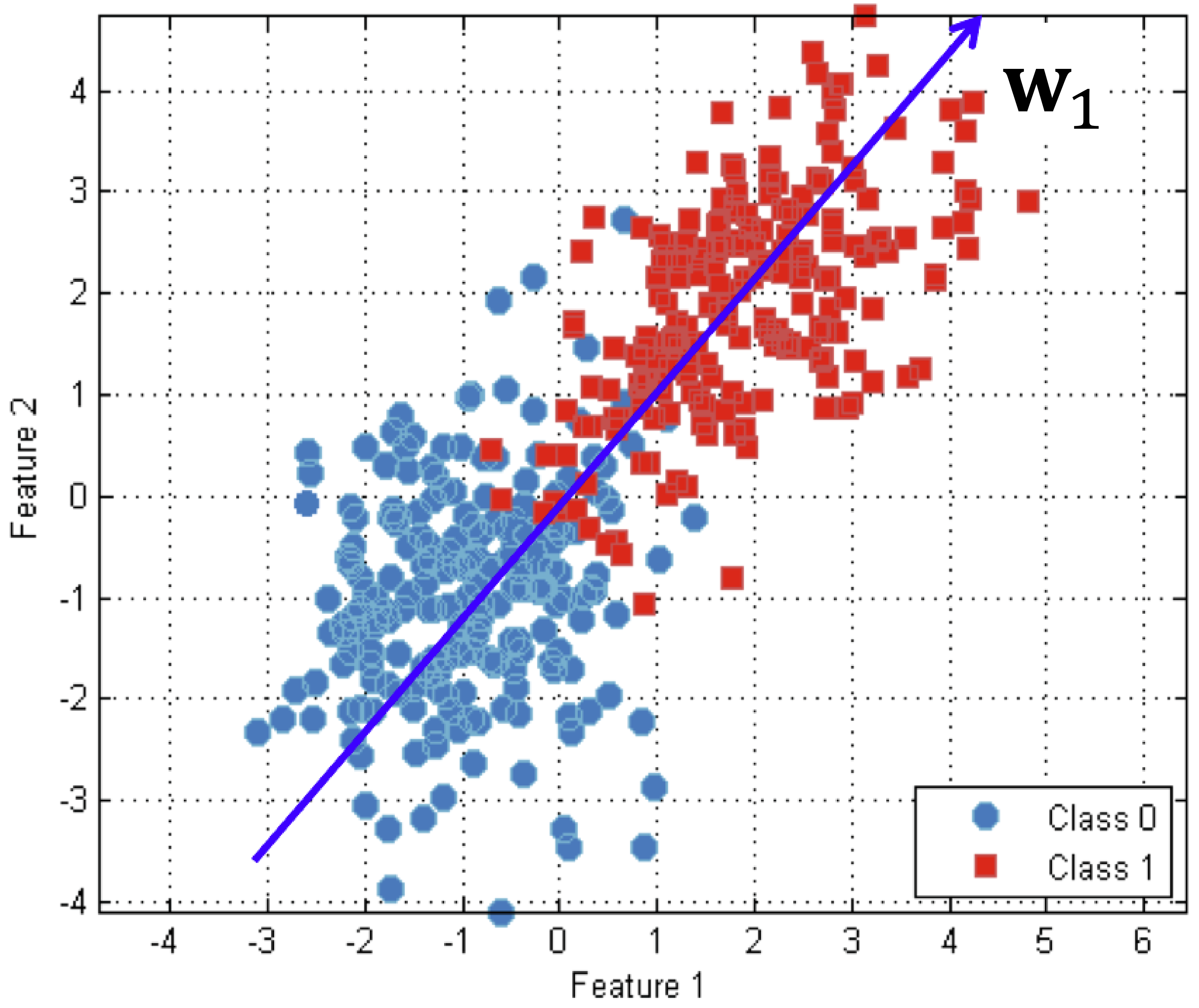
- Given a symmetric matrix $\mathbf{C} \in \mathbb{R}^{D \times D}$
- Find a vector $\mathbf{w} \in \mathbb{R}^{D}$ and $\|\mathbf{w}\|_2 = 1$
- Such that

$$\mathbf{C}\mathbf{w} = \lambda \mathbf{w}$$

- There will be multiple solutions of $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_D$ for its corresponding $\lambda_1, \lambda_2, \dots, \lambda_D$
- They are orthonormal:

$$\mathbf{w}_i^T \mathbf{w}_i = 1 \text{ and } \mathbf{w}_i^T \mathbf{w}_j = 0$$

Principal direction of the data



Variance in the principal direction

- Principal direction $\mathbf{w}$ satisfies:

$$\mathbf{C}\mathbf{w} = \lambda\mathbf{w} = \mathbf{w}\lambda$$

- Variance in the principal direction is

$$\mathbf{w}^T\mathbf{C}\mathbf{w} = \mathbf{w}^T\mathbf{w}\lambda$$

- Given that $\mathbf{w}^T\mathbf{w} = \|\mathbf{w}\|_\mathbf{2}^\mathbf{2} = 1$

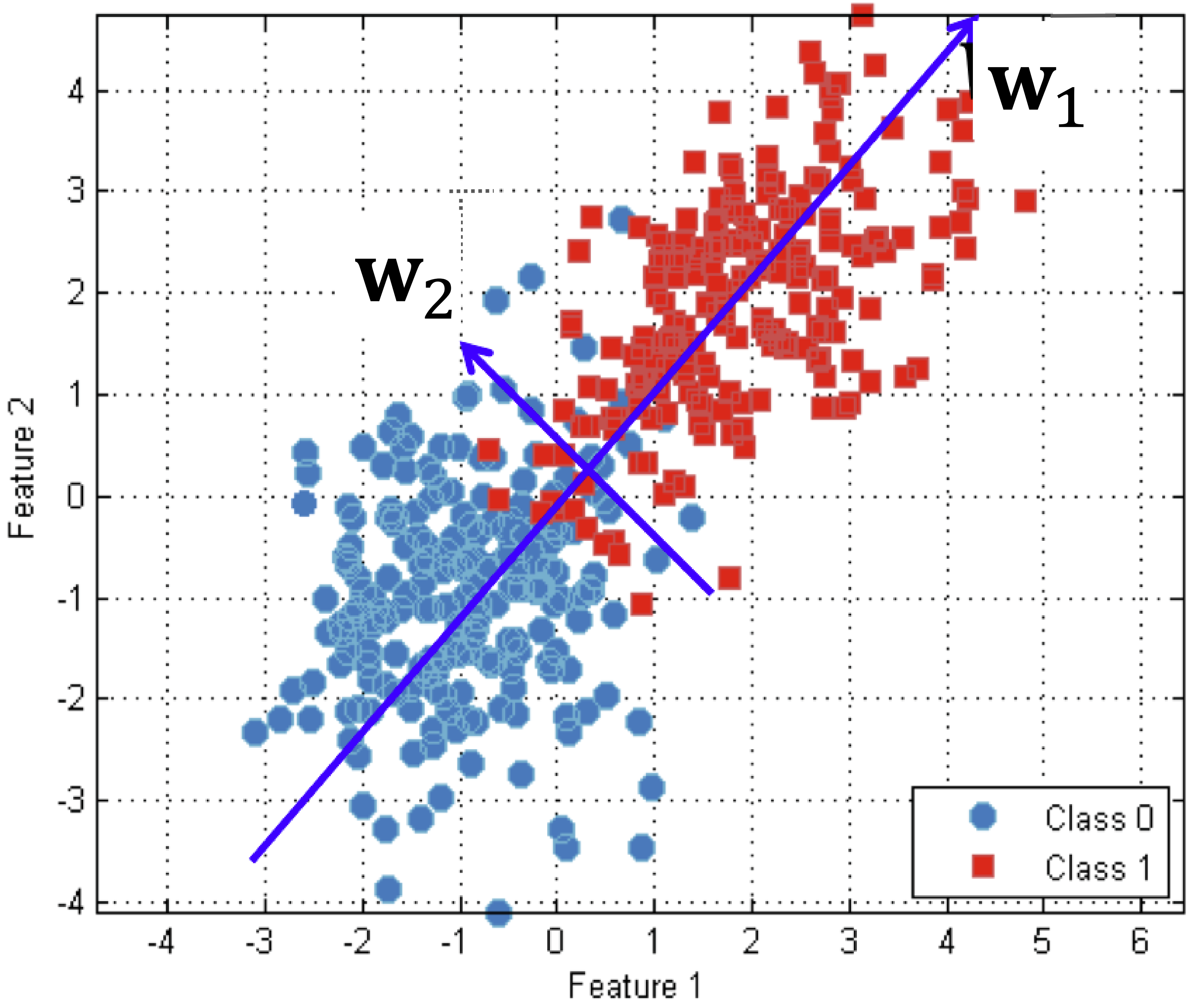
$$\mathbf{w}^T\mathbf{C}\mathbf{w} = \lambda$$

Multiple principal directions

- Directions $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_D$ has the largest variances but are orthogonal to each other

- Take the eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_D$ of $\mathbf{C}$ corresponding to:
  - The largest eigenvalue λ_1,
  - The second largest eigenvalue λ_2
  - …

Other 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 one, and
  - Points in the directions of the largest variance of the residual subspace

PCA algorithm

- Given N data points, $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^D$
- **Step 1:** estimate the mean and covariance matrix from data

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}_n \text{ and } \mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \boldsymbol{\mu})^T\,(\mathbf{x}_n - \boldsymbol{\mu})$$

- **Step 2:** take the eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_D$ of $\mathbf{C}$ corresponding to the largest eigenvalue λ_1, the second largest eigenvalue λ_2, …
- **Step 3:** Compute reduced representation

$$z_n = \begin{bmatrix} \mathbf{w}_1^T\left(\dfrac{\mathbf{x}_n - \boldsymbol{\mu}}{\sqrt{\lambda_1}}\right) \\ \mathbf{w}_2^T\left(\dfrac{\mathbf{x}_n - \boldsymbol{\mu}}{\sqrt{\lambda_2}}\right) \\ \cdots \\ \mathbf{w}_M^T\left(\dfrac{\mathbf{x}_n - \boldsymbol{\mu}}{\sqrt{\lambda_M}}\right) \end{bmatrix}$$

$$\lambda = \sigma^2$$

Outline

- Overview
- Principle component analysis: main idea
- The PCA algorithm
- PCA and SVD
- Summary

Singular value decomposition

- $\mathbf{X}_{N\times D}$, *N* is the number of dataset instances, *D* is the dimensionality of each instance (i.e. the number of features) and $\mathbf{X}$ is a centered matrix
- The singular value decomposition is given by

$$\mathbf{X} = \mathbf{U\Sigma V}^{\mathrm{T}}$$

$\mathbf{U}_{N\times N} \rightarrow unitary\ matrix \rightarrow \mathbf{UU}^{\mathrm{T}} = \mathbf{I}$
$\mathbf{\Sigma}_{N\times D} \rightarrow diagonal\ matrix$
$\mathbf{V}_{D\times D} \rightarrow unitary\ matrix \rightarrow \mathbf{VV}^{\mathrm{T}} = \mathbf{I}$

$$\underset{U}{\begin{bmatrix} u_{11} & \cdots & u_{1N} \\ \vdots & \ddots & \vdots \\ u_{N1} & \cdots & u_{NN} \end{bmatrix}_{N\times N}} \underset{\Sigma}{\begin{bmatrix} s_{11} & \cdots & 0 \\ 0 & \ddots & \vdots \\ \vdots & 0 & s_{DD} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}_{N\times D}} \underset{V^T}{\begin{bmatrix} v_{11} & \cdots & v_{1D} \\ \vdots & \ddots & \vdots \\ v_{D1} & \cdots & v_{DD} \end{bmatrix}_{D\times D}}$$

$$(D < N)$$

Covariance matrix and SVD

- Starting with the covariance matrix expression $\mathbf{C}_{D\times D} = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{N}$ and replacing $\mathbf{X} = \mathbf{U\Sigma V}^{\mathrm{T}}$ into the expression for the covariance, we obtain:

$$\mathbf{C} = \frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}}{N} \rightarrow \mathbf{C} = \frac{\mathbf{V\Sigma}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}}\mathbf{U\Sigma V}^{\mathrm{T}}}{N} = \frac{\mathbf{V\Sigma}^{2}\mathbf{V}^{\mathrm{T}}}{N}$$

- Multiplying the result by $\mathbf{V}$ on the right hand side:

$$\mathbf{CV} = \mathbf{V}\frac{\Sigma^2}{N}\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{V}\frac{\mathbf{\Sigma}^2}{N}$$

Covariance matrix and SVD

- According to the eigendecomposition definition $\mathbf{CV} = \mathbf{V\Lambda}$, therefore the eigenvalues of the covariance matrix are:

$$\lambda_i = \frac{\Sigma_\mathrm{i}^2}{N}$$

- λ_i: eigenvalue of $\mathbf{C}$ or covariance matrix
- Σ_i: singular value of $\mathbf{X}$ matrix

<u>So we can directly calculate eigenvalue of a covariance matrix by having the singular values of matrix $\mathbf{X}$</u>

SVD and PCA

- The $\mathbf{V}$ matrix corresponds to the eigenvectors of the covariance matrix (principal directions)

$$\lambda_i = \frac{\Sigma_{\mathrm{i}}^2}{N}$$

- To project the data matrix onto the principal directions, we compute:

$$\mathbf{X}_{proj} = \mathbf{XV} = \mathbf{U\Sigma}$$

Where X_{proj} consists of a linear combination of the original data

- We then truncate our projected matrix to the number of principal components M we would like to use.

SVD and PCA

Eigenvalues: $\lambda_i = \frac{\boldsymbol{\Sigma}_i^2}{N}$

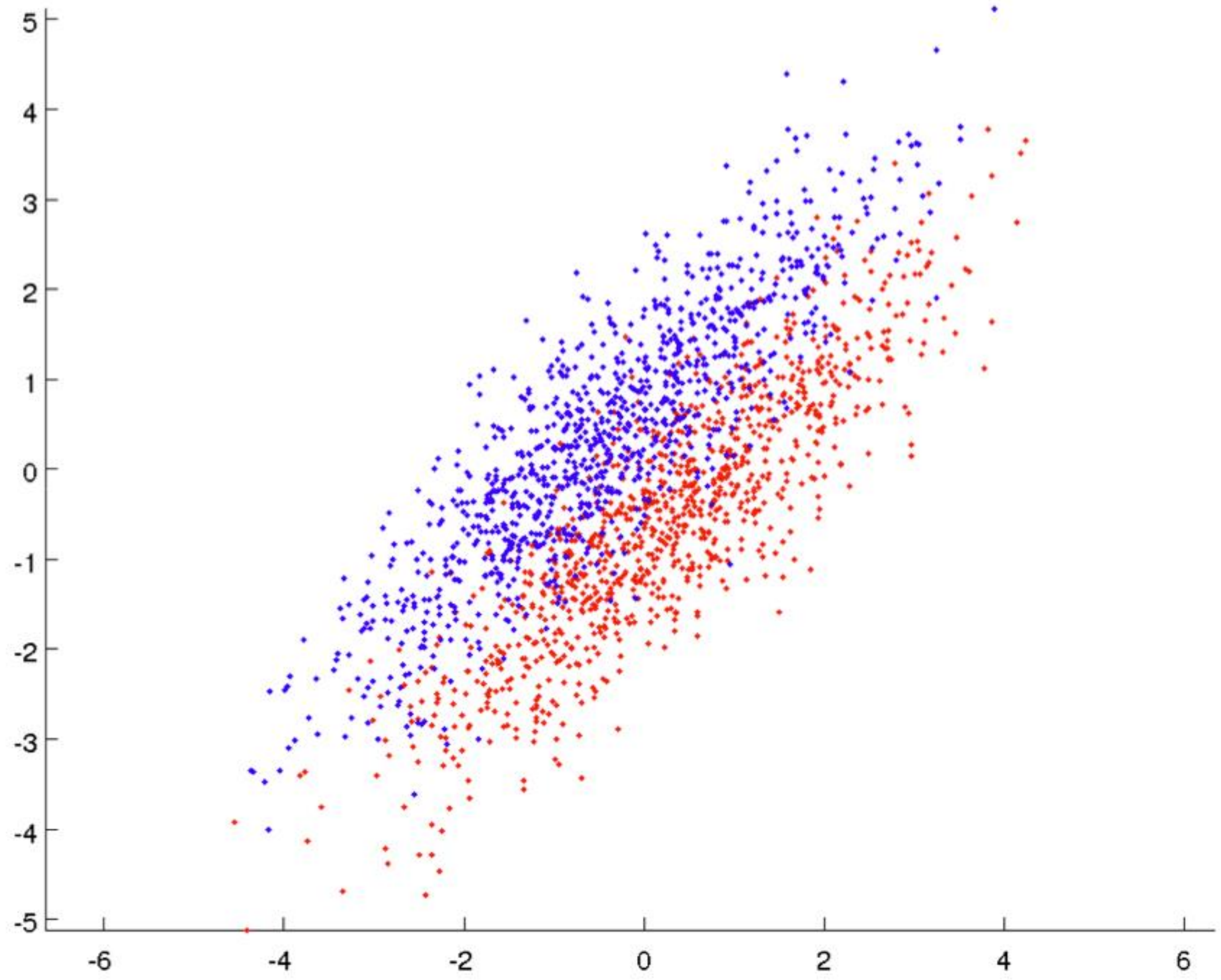
Eigenvectors (principal directions) $\mathbf{V}$

$$X = \underbrace{U\Sigma}_{\text{Principal components or projections on principal directions}} V^T$$

Principal components 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 $\mathbf{X}^T\mathbf{X}$ 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