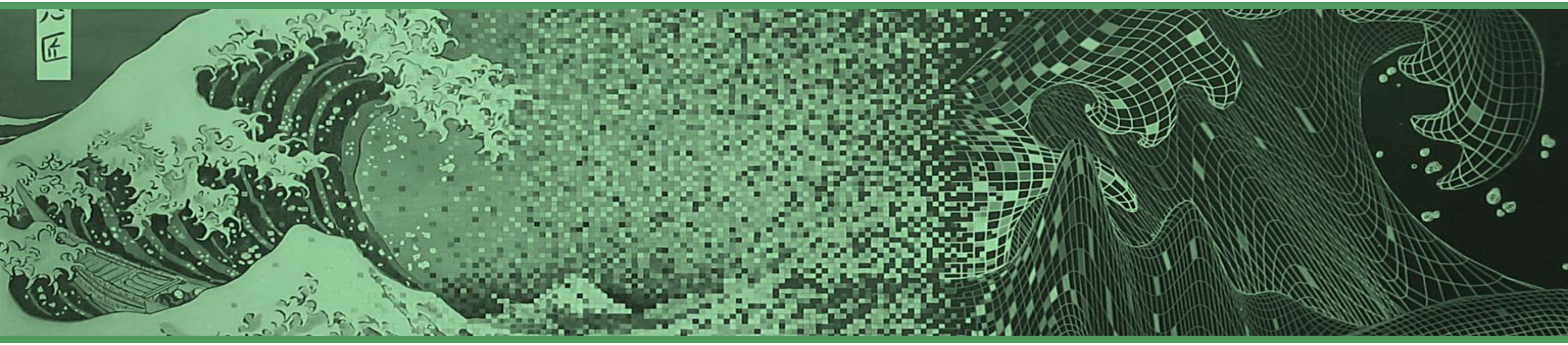
TÉCNICO LISBOA

Dimensionality reduction

Feature selection and PCA transformations



Aprendizagem 2023

Rui Henriques rmch@tecnico.ulisboa.pt
Andreas Wischert andreas.wichert@tecnico.ulisboa.pt

Outline

- **High-dimensional data**
- **Feature selection**
- **Feature extraction**
  - algebra essentials: eigenvalues and eigenvectors
  - KL transform
  - principal component analysis
  - additional notes
    - linear discriminant analysis
    - pseudoinverse
    - alternative approaches to dim reduction

Motivation

- At a first glimpse, increasing the number of variables should lead to better performance…
- In practice, the inclusion of more features can degrade performance (**curse of dimensionality**)
  - *challenges*: learning complexity and generalization difficulty (over/underfitting)
  - common definition of **high-dimensionality**: $|Y| \gg |X|$ (i.e. $m \gg n$)
- The number of training observations required increases **exponentially** with dimensionality
- How then can we learn in high-dimensional data spaces with a limited number of observations?
  - revise the learning approach
    - *example*: adequate distances in high-dimensional data spaces for lazy learning and clustering
  - **dimensionality reduction** $\Leftarrow$

Data domains with high-dimensionality

- **biological data**
  - gene expression (>20k genes)
  - molecular concentrations (metabolites, proteins…)
- **text** and **web content** data
- **social** behavioral data
- **healthcare** data (clinical records)
- **consumer data**
- **signal, audio, image** and **video** data

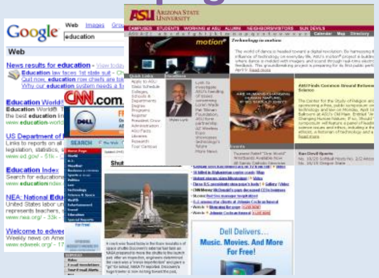
thousands of terms

documents

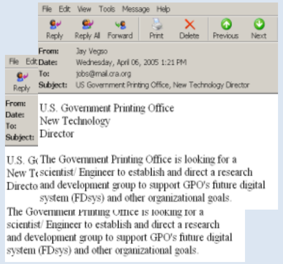
	t_1	t_2	………	t_m	C
d_1	12	0	………	6	sports
d_2	3	10	………	28	travel
⋮	⋮			⋮	⋮
d_n	0	11	………	16	jobs

Task: classify unlabeled documents
Challenge: thousands of terms
Solution: dimensionality reduction

web pages

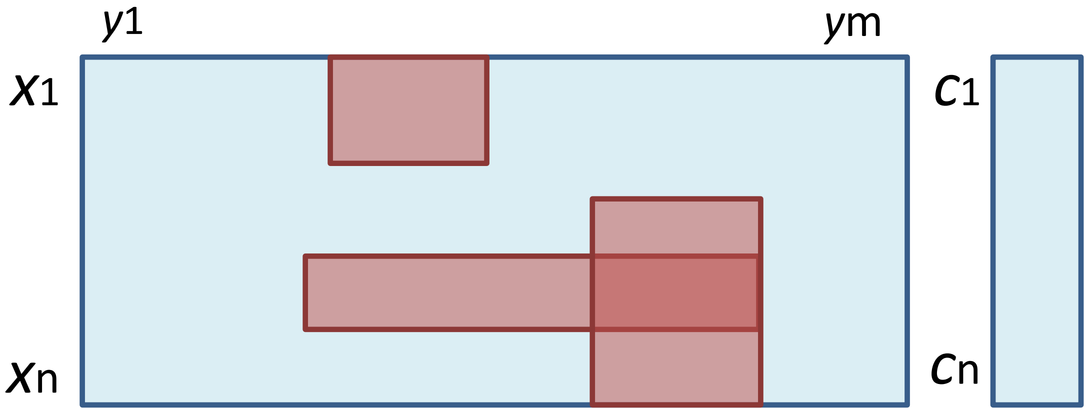


emails



Generalization: overfitting and underfitting risks

- **overfitting**
  - unability to discard non-informative and/or non-discriminative regions
  - incidence: <u>global learning</u>
    - e.g. naïve Bayes, neural networks, SVMs...



- **underfitting**
  - exclusion of informative or discriminative regions from the learning
  - incidence: <u>local learning</u>
    - e.g. decision trees, kNN, pattern mining...

Goals of dimensionality reduction

- Guide **supervised learning** (focus on discriminative regions)
- Guide **unsupervised learning** (focus on informative regions)
- **Visualization** (project high-dim data into interpretable low-dim data)
- **Data compression** (efficient storage and retrieval)
- **Noise removal** (denoising data)
- **Speed-up** learning
- Guarantee simplicity and comprehensibility of mined results
- Map **multimedia data** (image and signal data) into feature-based data

Outline

- High-dimensional data
- **Feature selection**
- Feature extraction
  - algebra essentials: eigenvalues and eigenvectors
  - KL transform
  - principal component analysis
  - additional notes
    - linear discriminant analysis
    - pseudoinverse
    - alternative approaches to dim reduction

Dimensionality reduction

- Project the m-dimensional points into a k-dimensional space ($k \ll m$)
  - preserve most of relevant information or structure from data
- Solve the learning problem in low dimensions
- Two common approaches
  - **feature selection** $\Leftarrow$
    - choosing a subset of all features

      $[y_1, y_2, \ldots, y_m] \longrightarrow [y_{i1}, y_{i2}, \ldots, y_{ik}]$
  - feature extraction
    - creating new features by combining existing ones

      $[y_1, y_2, \ldots, y_m] \longrightarrow [c_1, c_2, \ldots, c_k] = f([y_{i1}, y_{i2}, \ldots, y_{im}])$

Feature selection

- as a ***filter***: measure feature importance and select top-*k* features or above importance threshold
  - **unsupervised** settings
    - categorical features with **high entropy**
    - numeric features with **high variability**
  - **supervised** setting
    - **classification**, e.g. features with **high information gain**
    - **regression**, e.g. features with **high correlation**
    - check our former class on univariate data stances!
- as a ***wrapper***: assess learning performance with varying subsets of features
  - simplest way: to measure feature importance and test models on top-*k* features with varying *k*

Feature selection

- Example
  - **entropy** of a variable
  - variable-conditional entropy
  - **information gain**

$$H(y_j) = -\sum_{v \in y_j} P(v) \log_2 P(v)$$

$$H(z|y_j) = \sum_{v \in y_j} P(v) E(z|v)$$

$$IG(z|y_j) = H(z) - H(z|y_j)$$

	Hair	Height	Weight	Lotion	Result
i_1	1	2	1	0	1
i_2	1	3	2	1	0
i_3	2	1	2	1	0
i_4	1	1	2	0	1
i_5	3	2	3	0	1
i_6	2	3	3	0	0
i_7	2	2	3	0	0
i_8	1	1	1	1	0

$$\mathrm{rank}(\mathrm{hair}) = \mathrm{IG}(\mathrm{result}|\mathrm{hair}) = 0.45$$

$$\mathrm{rank}(\mathrm{height}) = \mathrm{IG}(\mathrm{result}|\mathrm{height}) = 0.26$$

hair variable has higher IG than height, hence is more important and less susceptible to removal

Outline

- High-dimensional data
- Feature selection
- **Feature extraction**
  - **algebra essentials**: eigenvalues and eigenvectors
  - KL transform
  - principal component analysis
  - additional notes
    - linear discriminant analysis
    - pseudoinverse
    - alternative approaches to dim reduction

Dimensionality reduction

- Project the m-dimensional points into a k-dimensional space ($k \ll m$)
  - preserve most of relevant information or structure from data
- Solve the learning problem in low dimensions
- Two common approaches
  - feature selection
    - choosing a subset of all features

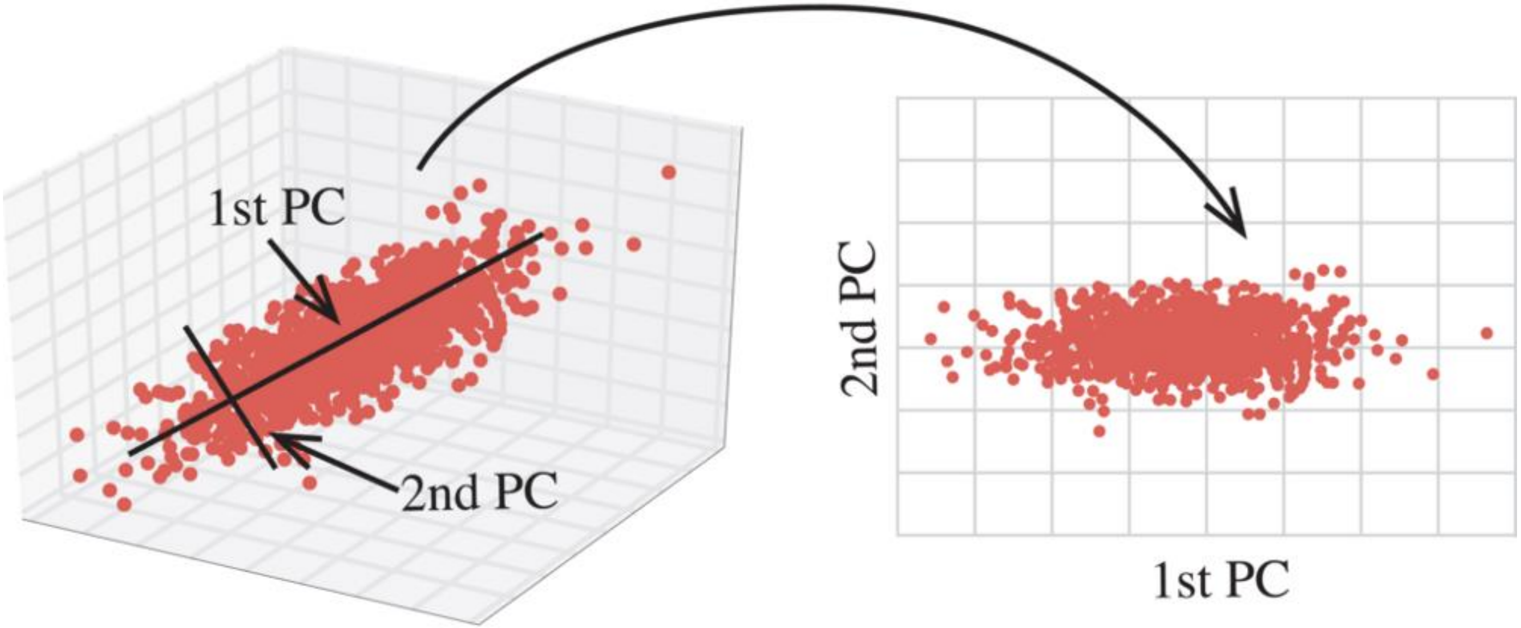
      $[y_1, y_2, \dots, y_m] \longrightarrow [y_{i1}, y_{i2}, \dots, y_{ik}]$
  - **feature extraction** $\Leftarrow$
    - creating new features by combining existing ones

      $[y_1, y_2, \dots, y_m] \longrightarrow [c_1, c_2, \dots, ck] = f([y_{i1}, y_{i2}, \dots, y_{im}])$

Feature extraction

- Find combinations of features that explain data
  - simple to compute and analytically tractable
- Classical approaches aim at finding a linear transformation
  - Goal: reduction that preserves as much information in data as possible (in a least-squares sense)
    - **Principal Component Analysis** (PCA)
- Simple extensions available to:
  - handle **non-linearity** (*kernel* trick)
  - sensitivity to **targets** (ensure new features yield discriminative power)
    - Goal: reduction that best separates the data (in a least-squares sense)
      - **Linear Discriminant Analysis** (LDA)

Space transformation



Axes of greater variance given by *eigenvectors* of *covariance matrix*

Covariance

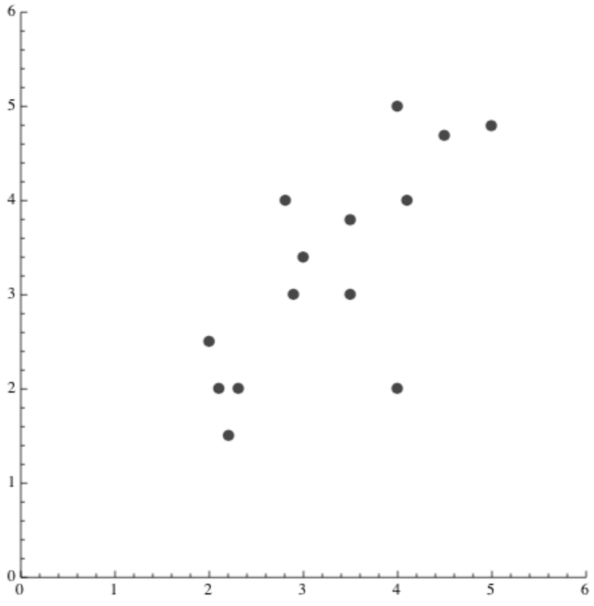
- The covariance matrix measures the tendency of two features, y_i and y_j, to vary in the same direction
  - covariance matrix C is symmetric and positive-definite
  - when normalizing covariance by their variances we obtain a correlation in [-1,1]
- Remember
  - sample covariance: $n-1$ in the denominator (Bessel's correction)

$$cov(y_1, y_2) = \frac{\sum_{i=1}^{n}(x_{1i} - \overline{y_1}) \cdot (x_{2i} - \overline{y_2})}{n-1}$$

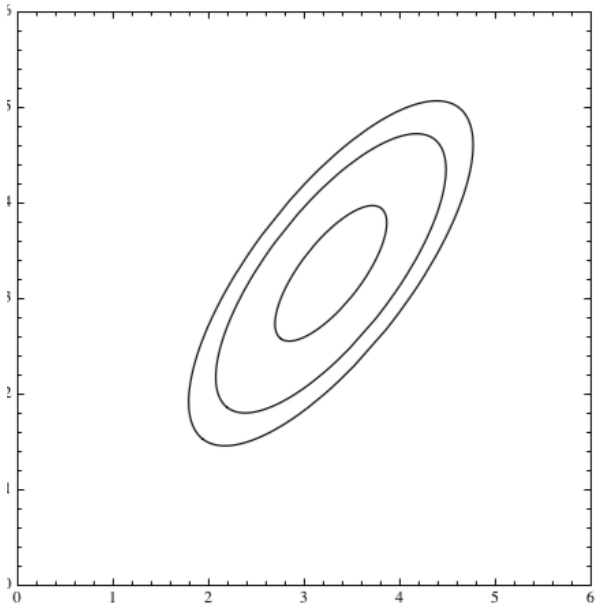
  - whole population: n is the denominator

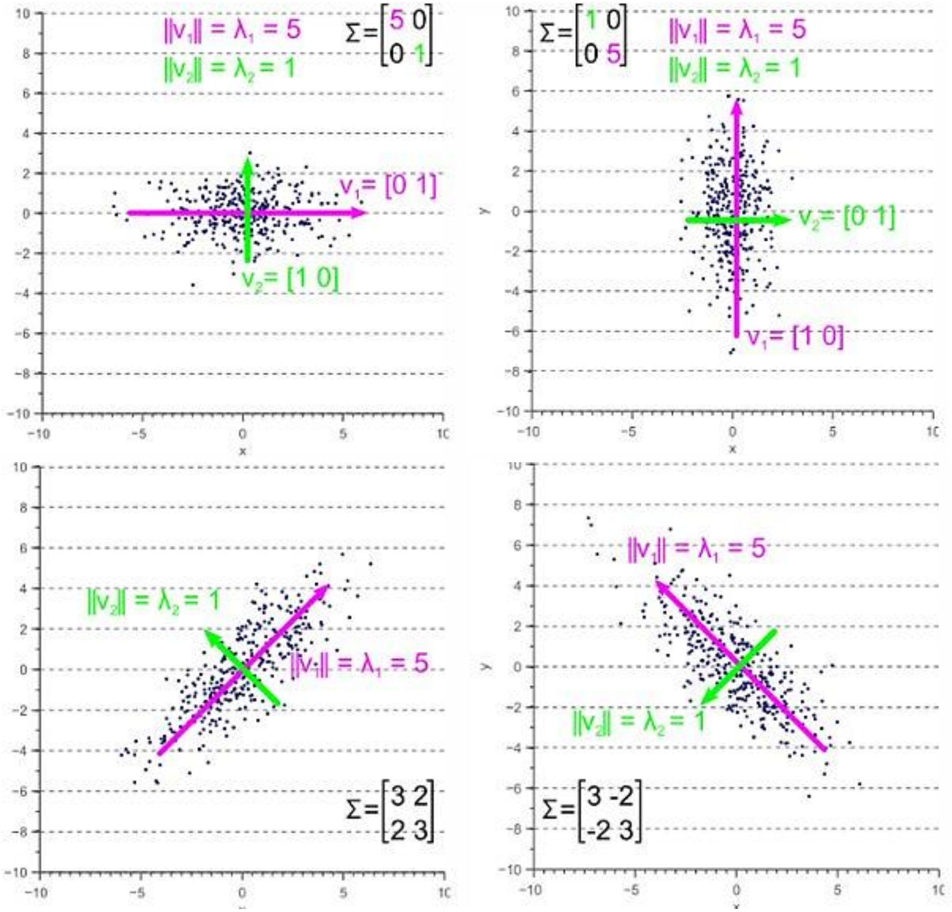
$$cov(y_1, y_2) = \frac{\sum_{i=1}^{n}(x_{1i} - \overline{y_1}) \cdot (x_{2i} - \overline{y_2})}{n}$$

Covariance



(a)





the covariance matrix of the data points defines the ellipses of equiprobability (defined by *eigenvectors* $\mathbf{v}$ and *eigenvalues* λ)

Eigenvalues and eigenvectors

- Let C be a $m \times m$ covariance matrix
- Vectors $\mathbf{v}$ having same direction as $C\mathbf{v}$ are called *eigenvectors*
  – eigenvectors define the linear composition of variables
- In the equation $\boxed{C\mathbf{v} = \lambda\mathbf{v}}$, λ is called an *eigenvalue* of A
- Example:

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4\begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$\mathbf{v} = [3\ 2]^T$ and $\lambda = 4$

meaning that data is described by $y_{new} = 3y_1 + 2y_2$

Eigenvalues and eigenvectors

- $C\mathbf{v} = \lambda\mathbf{v} \Leftrightarrow (C - \lambda I)\mathbf{v} = 0$
- Given A, how to calculate $\mathbf{v}$ and λ:
  - determine roots to $det(C - \lambda I) = 0$, roots are eigenvalues λ
  - solve $(C - \lambda I)\mathbf{v} = 0$ for each λ to obtain eigenvectors $\mathbf{v}$

y_1	y_2
-5.1	9.25
14.9	20.25
5.9	33.25
5.9	-30.75
…	…
-9.1	-10.75
-9.1	-21.75
5.9	19.25

$$C = \begin{pmatrix} 2 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}$$

Eigenvectors and eigenvalues:

$\mathbf{v}_1$=[0.91, 0.41], λ_1=2.36

$\mathbf{v}_2$=[-0.41, 0.91], $\boldsymbol{\lambda}_2$=0.23

$$\mathbf{x}_i = (0.91 \quad 0.41)\begin{pmatrix} x_{i1} \\ x_{i2} \end{pmatrix}$$

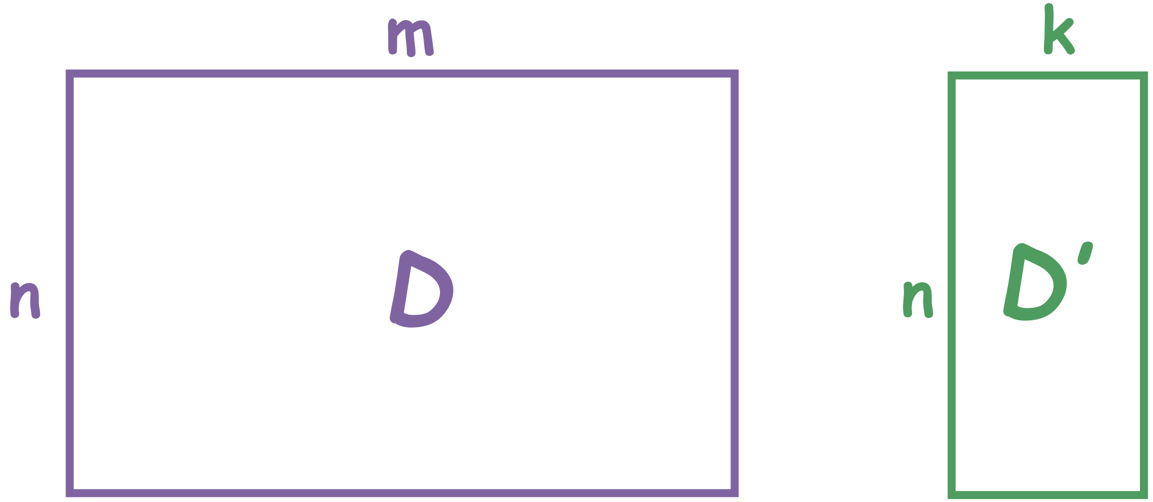
c_1
-0.8
21.9
19
-7.2
…
-12.7
-17.2
13.3

Outline

- High-dimensional data
- Feature selection
- **Feature extraction**
  - algebra essentials: eigenvalues and eigenvectors
  - **KL transform**
  - principal component analysis
  - additional notes
    - linear discriminant analysis
    - pseudoinverse
    - alternative approaches to dim reduction

Dimensionality reduction

- Map data with *m* variables into *k* variables without significant loss



- *Residual variation*: information in D not retained in D'
- Trade-off: k-dimensionality and interpretability *versus* information loss
  – the semantics of the variables are degraded in the reduced dataset D'

The Karhunen-Loève transform

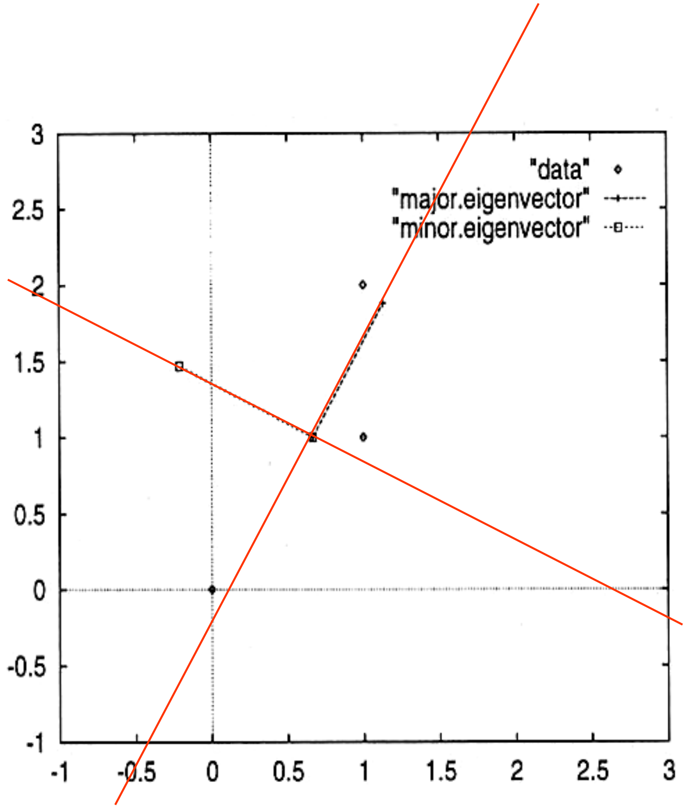
- *Intuition*: find the axis that shows the greatest variation and rotate that into this axis
- The Karhunen-Loève (KL) transform is a linear transform that maps possibly correlated variables into a set of values of linearly uncorrelated variables
  - centering data and computing covariance matrix
  - eigenvectors that minimize sum of square differences

$$D = \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}$$

$$D_{centered} = \begin{bmatrix} 1/3 & 1 \\ 1/3 & 0 \\ -2/3 & -1 \end{bmatrix} \text{ and } C = \begin{bmatrix} 2/3 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\lambda_1 = 2.53 \quad \mathbf{v}_1 = \begin{bmatrix} 0.47 \\ 0.88 \end{bmatrix}$$

$$\lambda_2 = 0.13 \quad \mathbf{v}_2 = \begin{bmatrix} -0.88 \\ 0.47 \end{bmatrix}$$



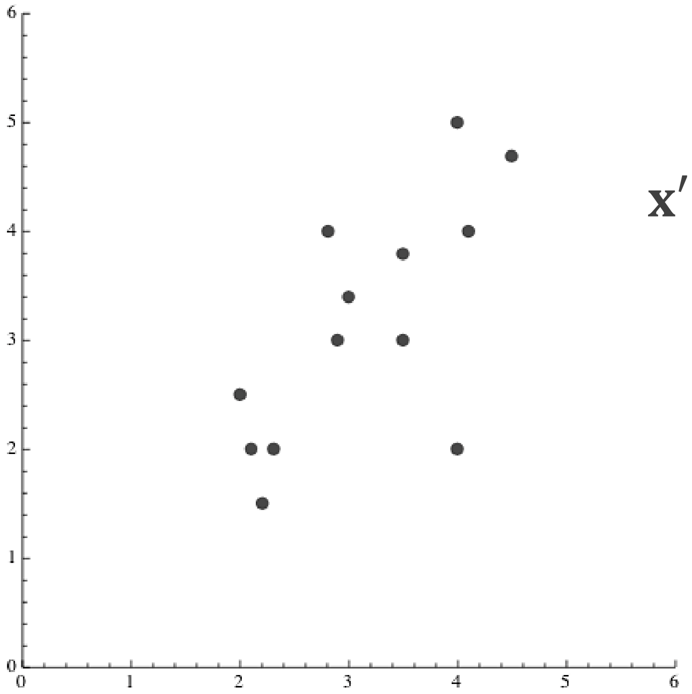
The Karhunen-Loève transform

- Transform defined by U matrix, orthonormal matrix of $m \times m$ dimension, i.e. $U^T \cdot U = I$
  - symmetric and positive definite that can be diagonalized

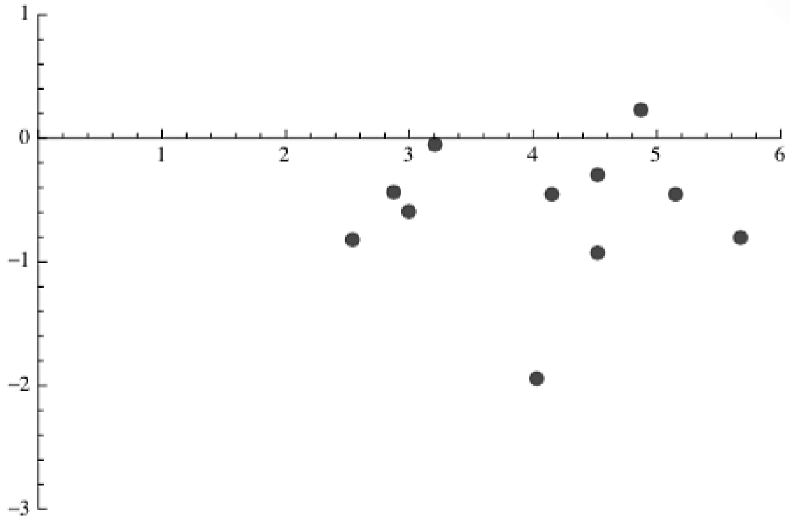
$$U^{-1}CU = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \lambda_m \end{pmatrix}$$

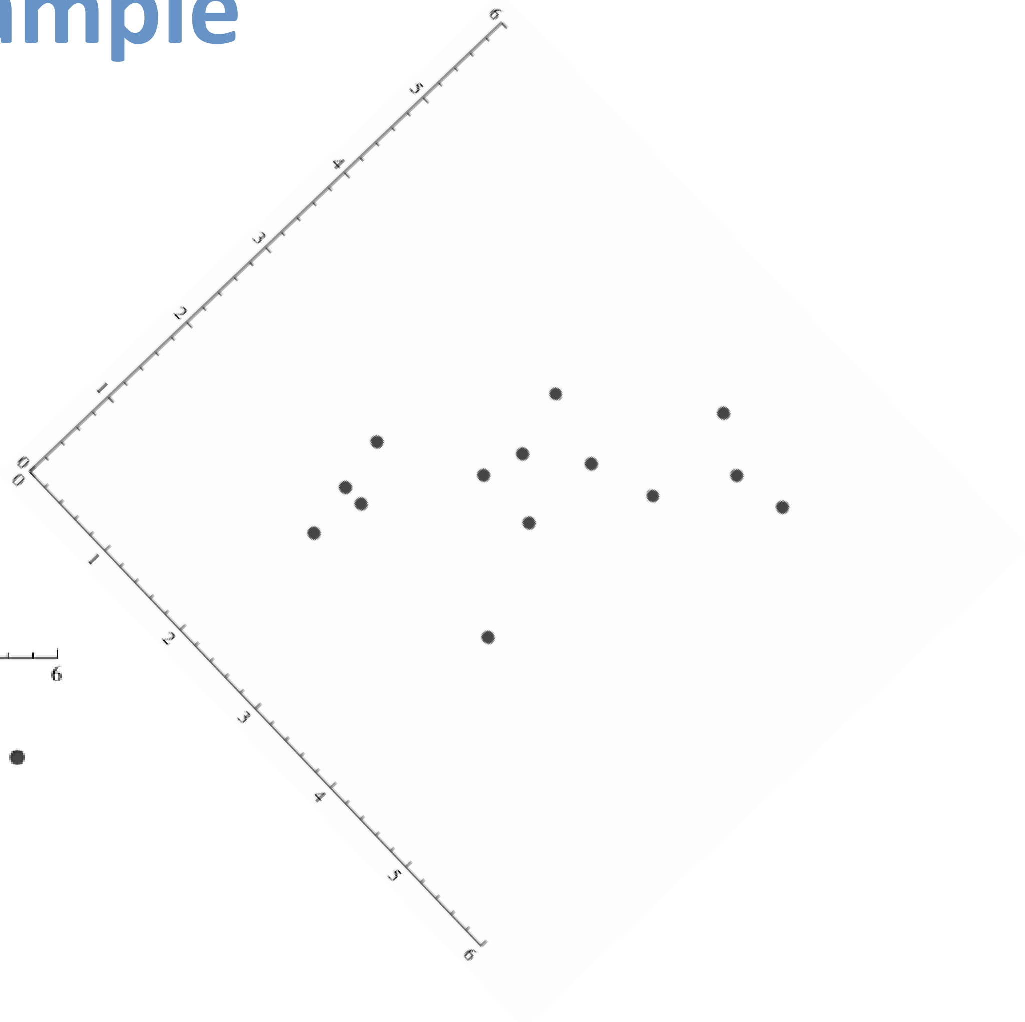
  - there are m eigenvalues and eigenvectors, $C\mathbf{v}_i = \lambda_i \mathbf{v}_i$
- The normalized eigenvectors define the orthonormal matrix U of dimension $m \times m$
  - each normalized eigenvector is a column
  - U defines the KL transform
    - KL transform rotates the coordinate system $\mathbf{x}' = U^T \cdot \mathbf{x}$

KL transform: example



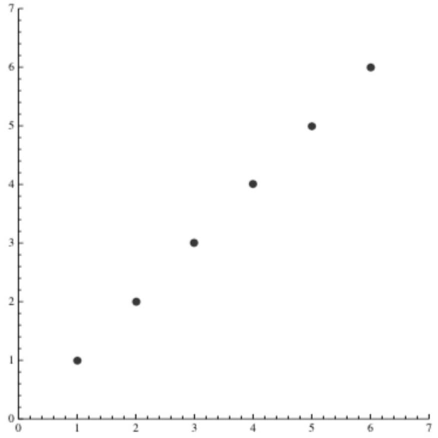
$$\mathbf{x}' = U^T \cdot \mathbf{x} = \begin{pmatrix} 0.61 & 0.79 \\ -0.79 & 0.61 \end{pmatrix} \cdot \mathbf{x}$$



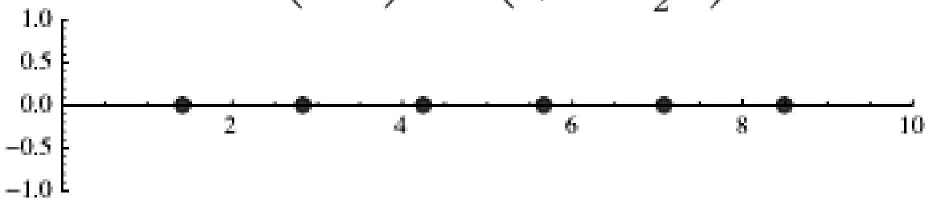


It rotates the system (the *points*) in such a way hat the new covariance matrix will be diagonal

KL transform: example



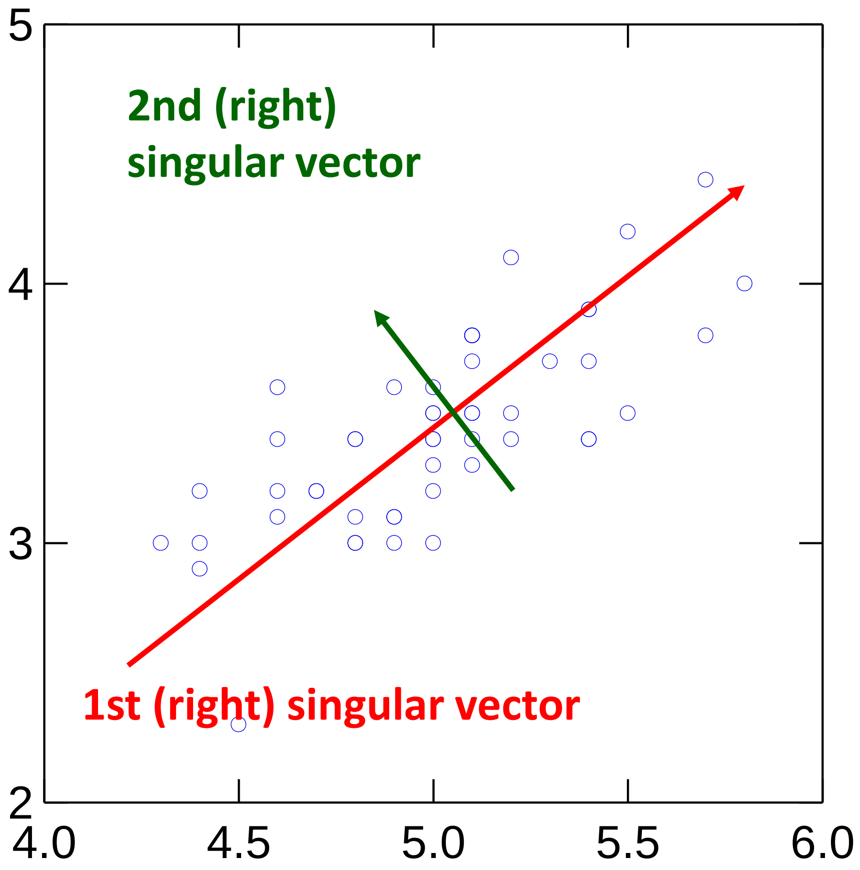
- Considering $D = \{(1,1), (2,2), (3,3), (4,4), (5,5), (6,6)\}$
  - the covariance matrix is $C = \begin{pmatrix} 3.5 & 3.5 \\ 3.5 & 3.5 \end{pmatrix}$
  - the two eigenvalues are $\lambda_1 = 7, \quad \lambda_2 = 0$
  - the two eigenvectors are $\mathbf{u}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \mathbf{u}_2 = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$
  - the matrix that describes the KL transform $U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \sqrt{2} \cdot \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$
  - let us transform the first observation $\begin{pmatrix} \sqrt{2} \\ 0 \end{pmatrix} = \begin{pmatrix} \sqrt{2} \cdot \frac{1+1}{2} \\ \sqrt{2} \cdot \frac{1-1}{2} \end{pmatrix} = \sqrt{2} \cdot \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}$
  - transformed data space



Outline

- High-dimensional data
- Feature selection
- **Feature extraction**
  - algebra essentials: eigenvalues and eigenvectors
  - KL transform
  - **principal component analysis**
  - additional notes
    - linear discriminant analysis
    - pseudoinverse
    - alternative approaches to dim reduction

Singular value decomposition



1^{st} singular vector:
direction of maximal variance
λ_1: how much of the data variance is explained by 1^{st} vector

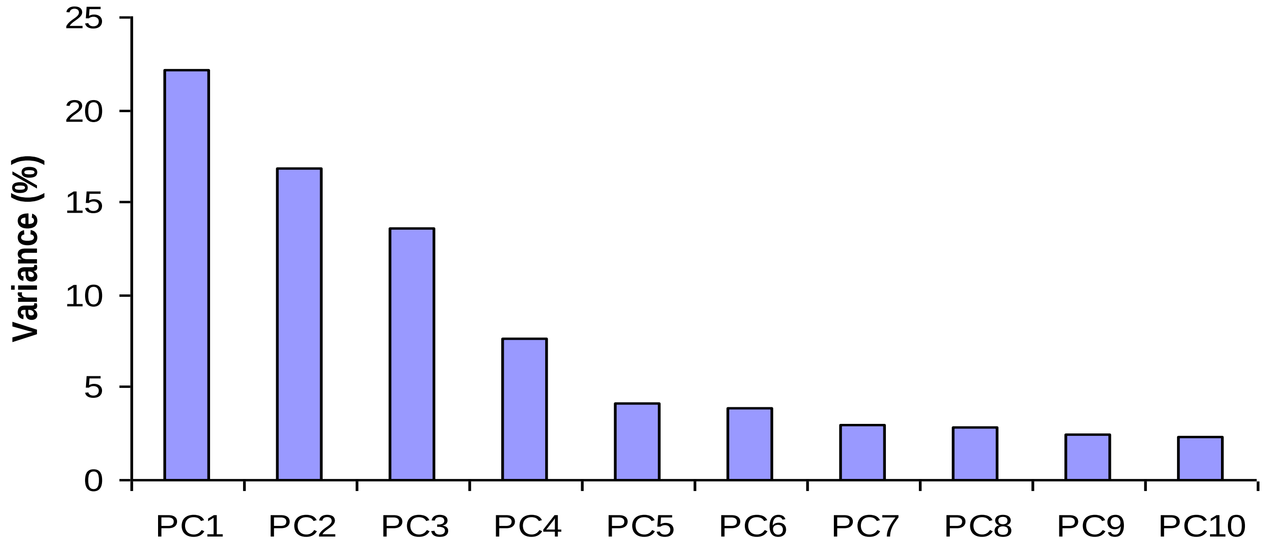
2^{nd} singular vector:
direction of maximal variance, after removing projection of 1^{st} vector
λ_2: how much of the data variance is explained by the 2^{nd} vector

…

m^{th} singular vector …
(until variance below threshold)

Principal components

- PCA is SVD done on centered data (singular vector/value = eigenvector/value)
- First component (PC1): highest eigenvalue (direction with greatest variation)
- Second component (PC2): direction with max variation orthogonal to PC1
- In general: *only few directions needed to capture most data variability*



Principal component analysis

- PCA projects data along the directions where the data varies most
- These directions are determined by the eigenvectors corresponding to the largest eigenvalues (magnitude defines the direction's variance)
  - reduction can imply information loss
  - SVD/PCA preserve as much information as possible by minimizing the reconstruction error
- Components (summary variables)
  - linear combinations of the original variables
  - uncorrelated with each other
  - the largest eigenvalues are called ***principal components***
  - the squares of the eigenvalues represent the variances along the eigenvectors

Principal component analysis

- The variance in the direction of the k^{th} singular vector (or principal component) is given by the singular value λ_k
  - singular values can be used to estimate how many components to keep
  - ***rule of thumb:*** keep enough to explain 85% of the variation

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{n} \lambda_j} \approx 0.85$$

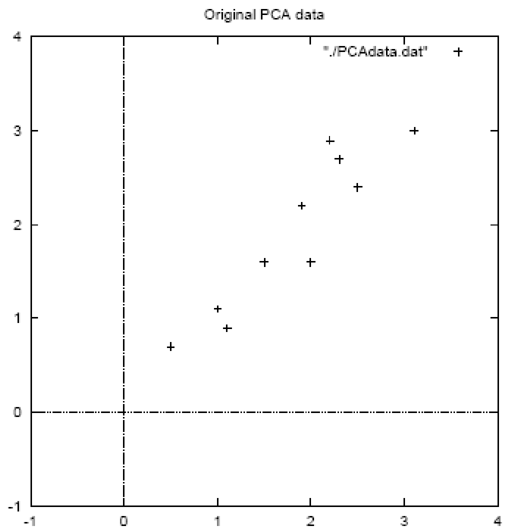
if $k = m$, we preserve 100% of the original variation

- Eigenvector $\mathbf{v}$ weights input variables to compose a new component $\mathbf{c}$
  - these absolute weights provide a view on the relevance of each input variable for component $\mathbf{c}$

Principal component analysis

- Revising **how**
  1. Compute the covariance matrix $m \times m$ (*scatter* of data)
  2. Compute eigenvalues, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$, and eigenvectors, $v_1, v_2, \ldots v_m$
  3. Keep the large k eigenvalues ($k \leq m$) and construct the transformed space
  4. Transform the dataset $D \rightarrow D'$

- **Exercise**: apply PCA on the following dataset



PCA example

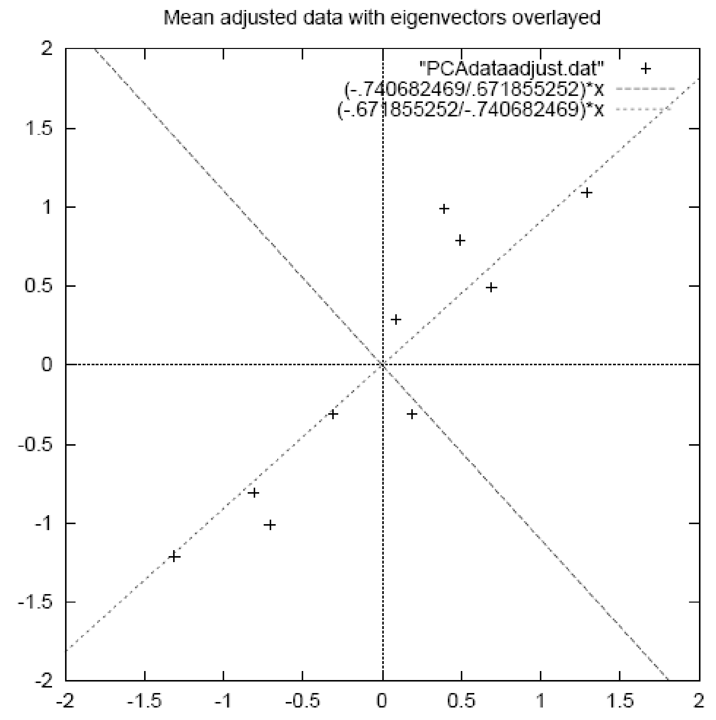
Optionally center data (removing mean)

Dataset D

$\mathbf{y}_1$	$\mathbf{y}_2$
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	1.6
1	1.1
1.5	1.6
1.1	0.9
mean 1.81	1.91

Centered dataset D'

$\mathbf{y}_1$	$\mathbf{y}_2$
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-0.31
-0.81	-0.81
-0.31	-0.31
-0.71	-1.01
mean 0	0



PCA example

1. Calculate the **covariance matrix**:

$$cov = \begin{array}{c c c} & y_1 & y_2 \\ & \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix} & \begin{array}{c} y_1 \\ y_2 \end{array} \end{array}$$

2. Calculate its (unit) **eigenvectors** and **eigenvalues**

$$eigenvalues = \begin{pmatrix} 0.049 \\ 1.284 \end{pmatrix}, \qquad eigenvectors = \begin{pmatrix} -0.735 & -0.678 \\ 0.678 & -0.735 \end{pmatrix}$$

3. Order eigenvectors by eigenvalue, highest to lowest and select top *p*

$$\mathbf{v}_1 = \begin{pmatrix} -0.6779 \\ -0.7352 \end{pmatrix} \quad \lambda_1 = 1.284 \qquad \mathbf{v}_2 = \begin{pmatrix} -0.7352 \\ 0.6779 \end{pmatrix} \quad \lambda_2 = .0491$$

… and construct the transformed feature vector

$$FeatureVector(k = 2) = \begin{pmatrix} -0.6779 & -0.7352 \\ -0.7352 & 0.6779 \end{pmatrix} \qquad FeatureVector(k = 1) = \begin{pmatrix} -0.6779 \\ -0.7352 \end{pmatrix}$$

PCA example

4. **Derive the new data set**

   $TransformedData = RowFeatureVector \times RowDataAdjust$

$$DataAdjusted = \begin{pmatrix} .69 & -1.31 & .39 & .09 & 1.29 & .49 & .19 & -.81 & -.31 & -.71 \\ .49 & -1.21 & .99 & .29 & 1.09 & 79 & -.31 & -.81 & -.31 & -1.01 \end{pmatrix}$$

$$FeatureVector(p = 2)^T = \begin{pmatrix} -.6779 & -.7352 \\ -.7352 & .6779 \end{pmatrix}$$

$$FeatureVector(p = 1)^T = (-.6779 \quad -.7352)$$

Transformed Data= $p = 2$

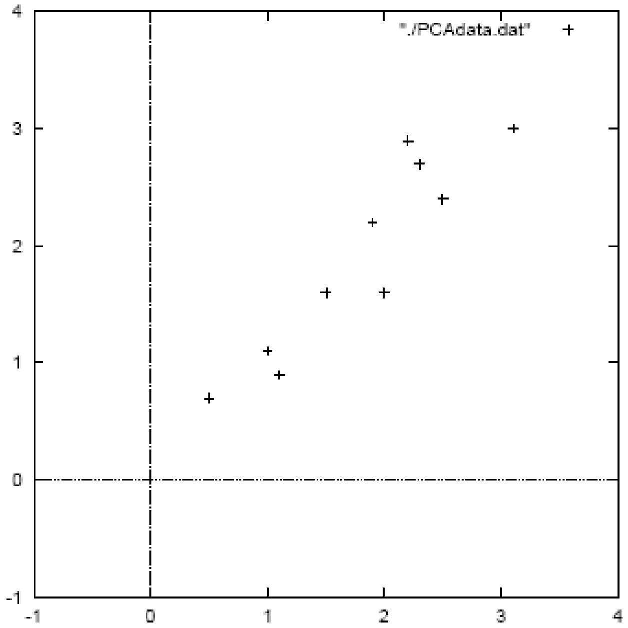
c_1	c_2
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

Transformed Data= $p = 1$

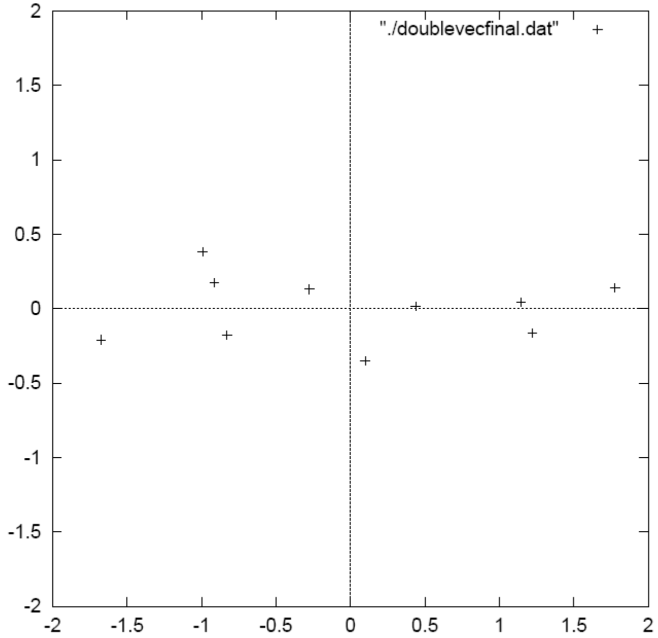
c_2
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

PCA example

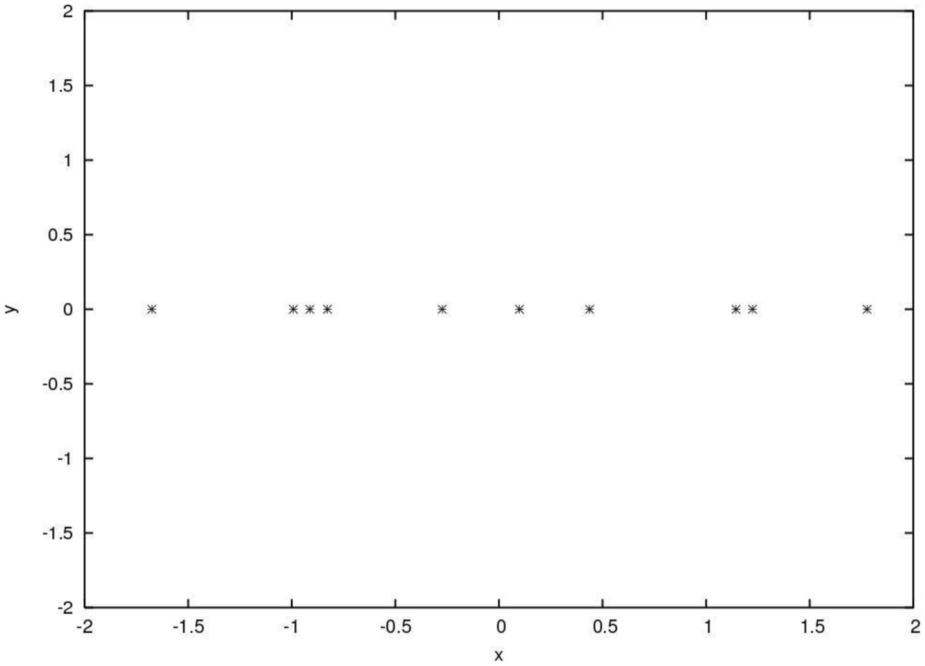
Original data



KL rotation
Data transformed with
k=2 eigenvectors



PCA $(k = 1)$
Data transformed with
k=1 eigenvector



(*uncovered*) Reconstruction error

- There is no information loss incurred in KL rotation ($k = m$)
- PCA minimizes the reconstruction error: $\|\mathbf{x} - \hat{\mathbf{x}}\|$
- It can be shown that the reconstruction error is: $error = 1/2 \sum_{i=k+1}^{m} \lambda_i$

- Example from previous slide:
  - using 2 components: recovery error = 0 (from previous slide)
  - using 1 component

y_1	y_2	y'_1	y'_2	$y^{\star}_1$	$y^{\star}_2$
2.5	2.4	0.56	0.61	2.4	2.5
0.5	0.7	-1.20	-1.31	0.6	0.6
2.2	2.9	0.67	0.73	2.5	2.6
1.9	2.2	0.19	0.20	2.0	2.1
3.1	3.0	1.14	1.23	2.9	3.1
2.3	2.7	0.62	0.67	2.4	2.6
2	1.6	-0.07	-0.07	1.7	1.8
1	1.1	-0.78	-0.84	1.0	1.1
1.5	1.6	-0.30	-0.32	1.5	1.6
1.1	0.9	-0.83	-0.90	1.0	1.0

$$error = 0.245 = \frac{0.49}{2} = \frac{\lambda}{2}$$

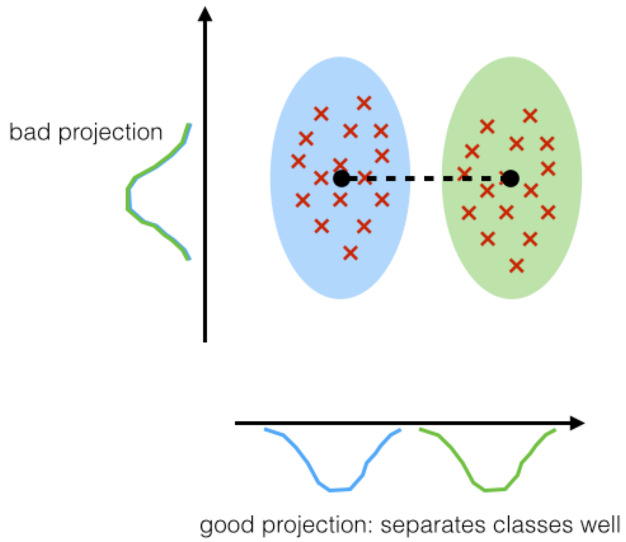
DataRecovered = (*FeatureVector*(p=1) x *TransformedData*) + *OriginalMean*

Outline

- High-dimensional data
- Feature selection
- **Feature extraction**
  - algebra essentials: eigenvalues and eigenvectors
  - KL transform
  - principal component analysis
  - additional notes
    - **linear discriminant analysis**
    - pseudoinverse
    - alternative approaches to dim reduction

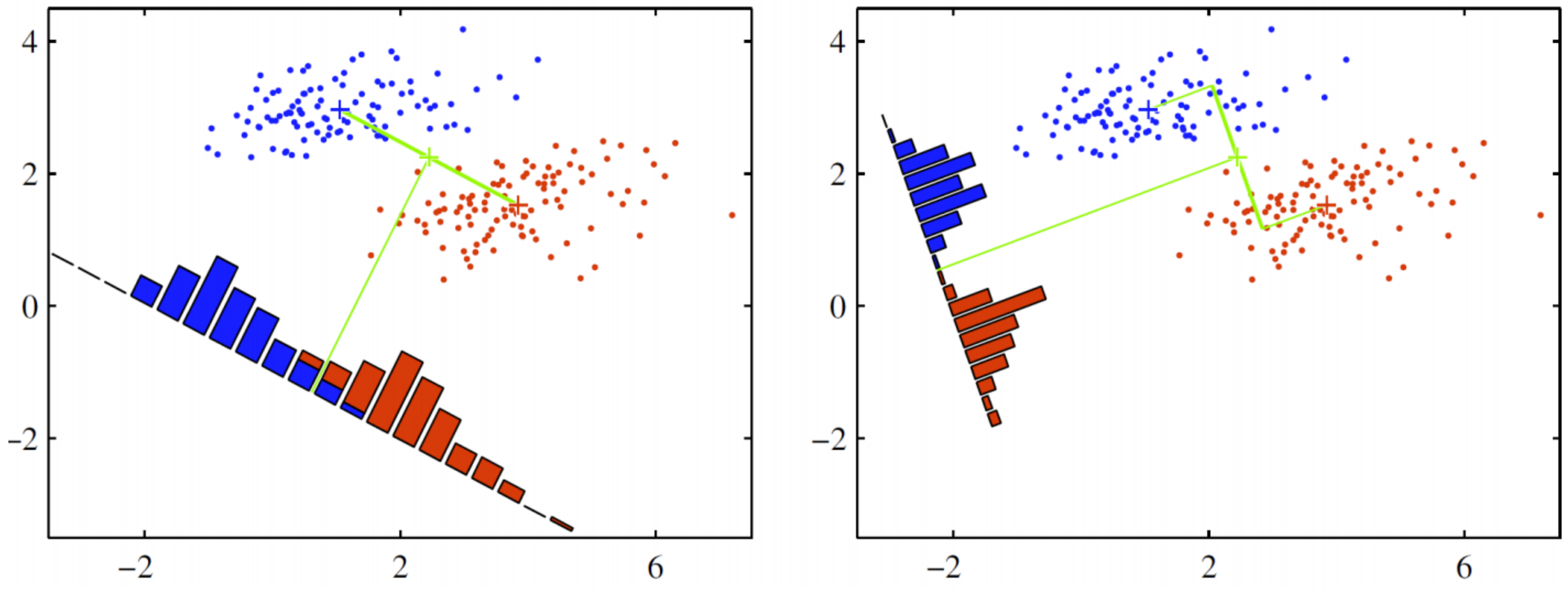
Linear discriminant analysis (LDA)

- Challenges of PCA in supervised settings?
  - Reduction does not consider impact on the ability to discriminate output variables
- Goal: data transformation guaranteeing class separation
- Principle: pick a new dimension that
  - **maximize separation between means of projected classes**
  - **minimize variance for the observations within each class**



- Solution: **LDA**
  - eigenvectors based on between-class and within-class covariance matrices

Linear discriminant analysis (LDA)



Outline

- High-dimensional data
- Feature selection
- **Feature extraction**
  - algebra essentials: eigenvalues and eigenvectors
  - KL transform
  - principal component analysis
  - additional notes
    - linear discriminant analysis
    - **pseudoinverse**
    - alternative approaches to dim reduction

(*uncovered*) **Pseudoinverse**

- At different moments of our journey we detect the need to invert matrices
  – e.g. linear regression closed-form solution
- *Problem*: not every matrix is invertible
- *Solution*: compute the pseudoinverse of D, i.e. $D^\dagger$, using SVD principles
  – so that $D^\dagger$is a right inverse, i.e. $D^\dagger D = I$

$$D^\dagger = U \cdot S' \cdot V^T$$

$$\underset{m \times n}{\overset{D^\dagger}{\begin{pmatrix} x_{11} & x_{12} & & x_{1n} \\ & & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}}} = \underset{m \times m}{\overset{\mathrm{U}}{\begin{pmatrix} u_{11} & & u_{m1} \\ & \ddots & \\ u_{1m} & & u_{mm} \end{pmatrix}}} \underset{m \times n}{\overset{\mathrm{S}}{\begin{pmatrix} \sigma_1 & & & 0 \\ & \ddots \; \sigma_r & & \\ & & \ddots & \\ 0 & & & 0 \end{pmatrix}}} \underset{n \times n}{\overset{\mathrm{V^T}}{\begin{pmatrix} v_{11} & & v_{1n} \\ & \ddots & \\ v_{n1} & & v_{nn} \end{pmatrix}}}$$

where U and V are orthogonal projections from $D^T D$ and DD^T

and S’ are the reciprocal non-zero elements from S

(*uncovered*) Pseudoinverse: example

$$D = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

$$DD^T = \begin{pmatrix} 17 & 8 \\ 8 & 17 \end{pmatrix}$$

$$\lambda_1 = 25, \lambda_2 = 9$$

$$\mathbf{u}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \mathbf{u}_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$D^T D = \begin{pmatrix} 13 & 12 & 2 \\ 12 & 13 & -2 \\ 2 & -2 & 8 \end{pmatrix}$$

$$\lambda_1 = 25, \lambda_2 = 9, \lambda_3 = 0$$

$$\mathbf{v}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}, \mathbf{v}_2 = \begin{pmatrix} {}^{1}\!/\!\sqrt{18} \\ -{}^{1}\!/\!\sqrt{18} \\ {}^{4}\!/\!\sqrt{18} \end{pmatrix}, \mathbf{v}_3 = \begin{pmatrix} 2/3 \\ -2/3 \\ -1/3 \end{pmatrix}$$

$$D^\dagger = USV^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} {}^{1}\!/\!\sqrt{2} & {}^{1}\!/\!\sqrt{2} & 0 \\ {}^{1}\!/\!\sqrt{18} & -{}^{1}\!/\!\sqrt{18} & {}^{4}\!/\!\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{pmatrix}$$

Outline

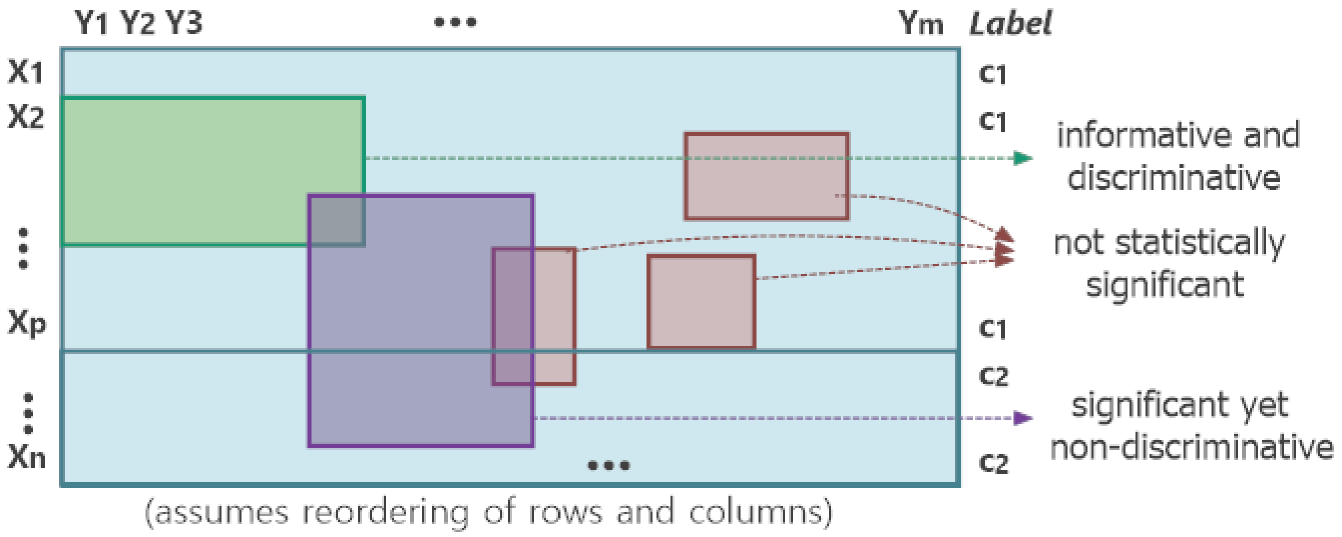
- High-dimensional data
- Feature selection
- **Feature extraction**
  - algebra essentials: eigenvalues and eigenvectors
  - KL transform
  - principal component analysis
  - additional notes
    - linear discriminant analysis
    - pseudoinverse
    - **alternative approaches to dim reduction**

How? Dimensionality reduction

1. **Feature selection**
2. **Feature extraction**/transformation
   - KL transformation and principal component analysis
   - linear discriminant analysis
3. **Sparse kernels** and **regularization** applied in parametric models to exclude non-relevant parameters
   - neural networks (excluding interactions)
   - regression approaches (sparse hyperplanes)
   - support vector machines
4. **Subspace selection** to jointly select variables and observations
   - pattern mining, decision trees and random forests
   - associative classifiers and decision tables

Alternative dimensionality reduction: subspace selection

- Addresses limitations of feature selection: single space → multiple compact spaces
- While…
  - minimizes overfitting: remove uninformative regions (focus on informative/discriminative subspaces)
  - minimizes underfitting: mine all relevant subspaces



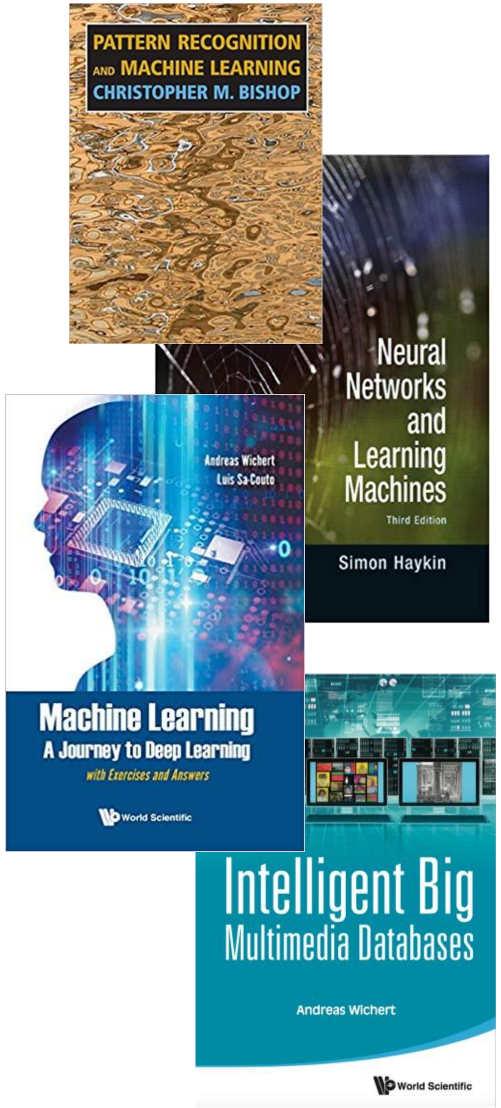
Outline

- **High-dimensional data**
- **Feature selection**
- **Feature extraction**
  - algebra essentials: eigenvalues and eigenvectors
  - KL transform
  - principal component analysis
  - additional notes
    - linear discriminant analysis
    - pseudoinverse
    - alternative approaches to dim reduction

Summary

- **High-dimensional data analysis** susceptible to under- and **overfitting risks**
- **Correlation** and **information theoretic measures** for (un)supervised **feature selection**
- **Feature transformation** tackles limitations of feature selection
- **Eigenvalue analysis** algorithms (KL, PCA, SVD) project data into orthogonal axes where data varies the most
- **Principal components**: linear combination of original features
- Dimensionality can be fixed in accordance with error-tolerance (explained variability)
- **Kernels** can be placed to handle non-linear data
- Transformations ca be evaluated by assessing models before-and-after reduction, and by plotting learning curves of **reconstruction error**

Literature



- C. Bishop, Pattern Recognition and Machine Learning, Springer 2006
  – Chapter 12
- S. Haykin, Neural Networks and Learning Machine, Pearson 2008
  – Chapter 10
- A. Wichert, L. Sa-Couto, Machine Learning – A Journey to Deep Learning, World Scientific, 2021
  – Chapter 15
- A. Wichert, Intelligent Big Multimedia Databases,, World Scientific, 2015
  – *Chapter 3, Section 3.3*

Thank You

rmch@tecnico.ulisboa.pt
andreas.wichert@tecnico.ulisboa.pt