Outline I

Supervised and Unsupervised Classification of Hyperspectral Images

An Introduction to the use of spatial information and of sparsity.

May 8, 2023

Gabriel Dauphin

https://www-l2ti.univ-paris13.fr/~dauphin/HIP.htm

Gabriel.dauphin@univ-paris13.fr
(please mention HIP in the subject of emails).

Outline II

Content

Intent of the lessons

- Objective $\leftrightarrow$ Recognize similarities
- Mathematical framework
- Simple implementation
- Graphics

Out of the scope

- State-of-the-art
- Efficient algorithms
- Multiclass
- Ensemble classifiers
- Overfitting

Table of Contents

What is an image

$$\left(f_{mn}^{R}, f_{mn}^{G}, f_{mn}^{B}\right)$$

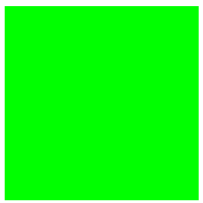
Exercise 1

What image is this showing?

```
R=[1;1;0]; G=[0.5;1;1]; B=[0;1;0];
im=cat(3,R,G,B),
figure(1); imshow(im);
```

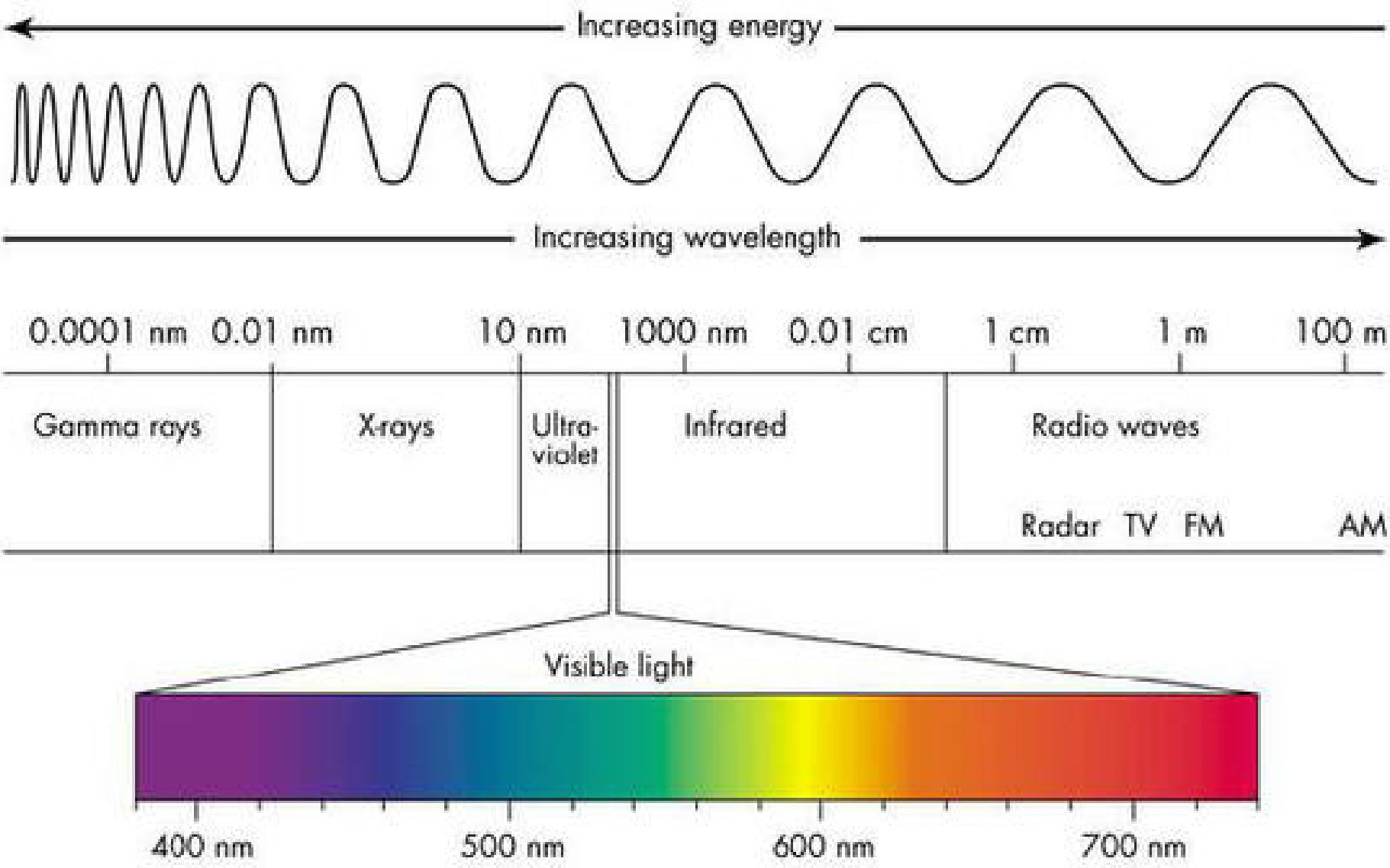
Answer to exercise 1

```
octave:3> im
im =

ans(:,:,1) =

   1
   1
   0

ans(:,:,2) =

   0.50000
   1.00000
   1.00000

ans(:,:,3) =

   0
   1
   0
```

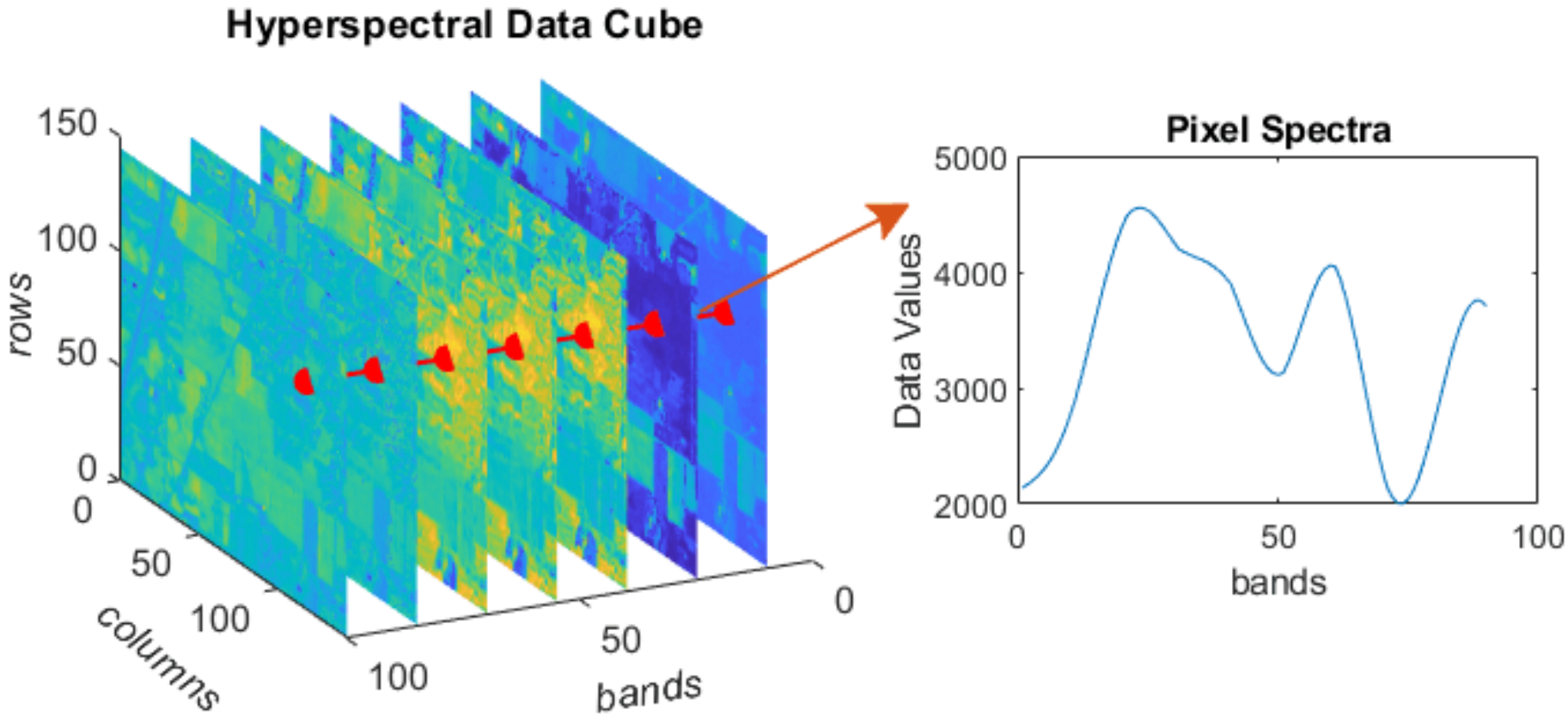


```
R=[1;1;0]; G=[0.5;1;1]; B=[0;1;0];
im=cat(3,R,G,B),
figure(1); imshow(im);
```

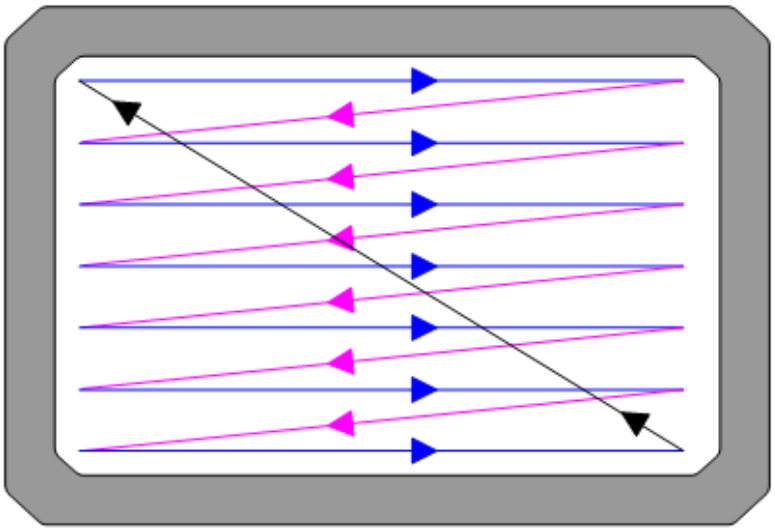
Wavelengths



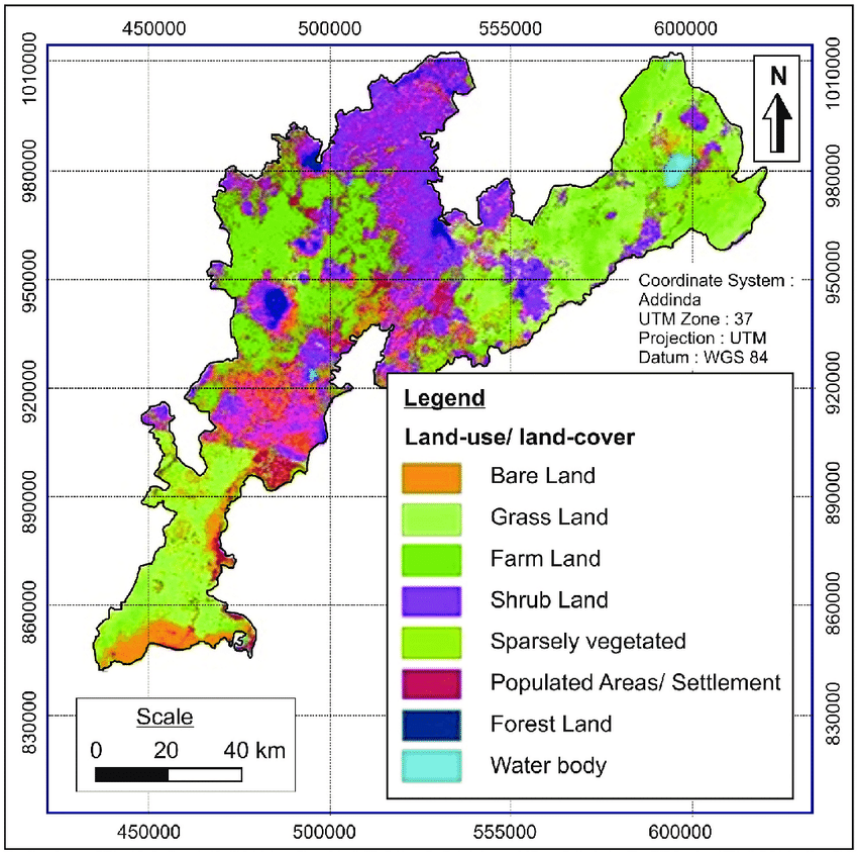
Hyperspectral image



Raster scanning order



- Feature space $\mathbf{x} \in \mathbb{R}^F$
- Input matrix
$$\mathbf{x} = [x_{nf}]_{n,f}$$
- Sample, instance or record $\mathbf{x}_n$
- Set of samples
$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$



- Classes $y_n \in \{0 \dots C-1\}$.
- Binary classification problem $C = 2$, $y_n \in \{0, 1\}$.
- Label column vector.

$$Y = [y_n]_n$$

Proximity in the feature space means
Labels are more **likely** to be the same

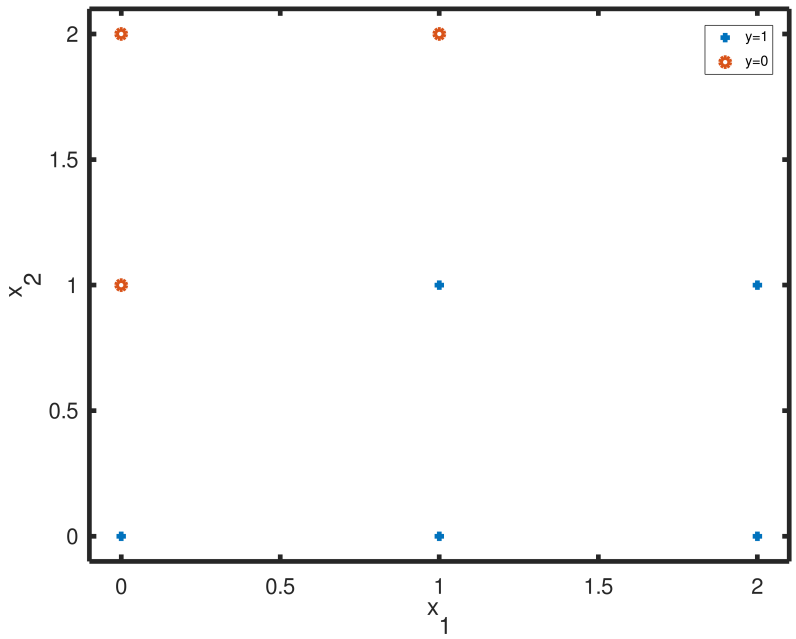
Exercise 2

Draw and code with Octave the scatter plot of the following dataset

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 2 & 0 \\ 2 & 1 \\ 2 & 2 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Answer to exercise 2

```
X=zeros(9,2);
X(:,1)=[0 0 0 1 1 1 2 2 2]';
X(:,2)=[0 1 2 0 1 2 0 1 2]';
Y=[1 0 0 1 1 0 1 1 1]';
ind1=find(Y==1);
ind0=find(Y==0);
figure(1); plot(X(ind1,1),...
X(ind1,2),'+',...
  'LineWidth',3,...
  X(ind0,1),...
X(ind0,2),'o',...
'LineWidth',3);
legend('y=1','y=0');
axis([-0.1 2.1 -0.1 2.1]);
```



Reordering a dataset

Exercise 3

Considering a binary dataset $(\mathbf{X}, Y)$ *composed of* $N = 3$ *samples belonging to a feature space of size* F*, and considering a matrix* T *of size* 3×3 *defined as*

$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

show that $(T\mathbf{X}, TY)$ *is the same dataset.*

Left multiplication

Left multiplication acts on the samples, whereas right multiplication acts on the features.

Answer to exercise 3

$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Denoting $\mathbf{x}_n = [x_{n1}, x_{n2}, x_{n3}]$ the rows of $\mathbf{x}$ and y_n the components of Y, we see that

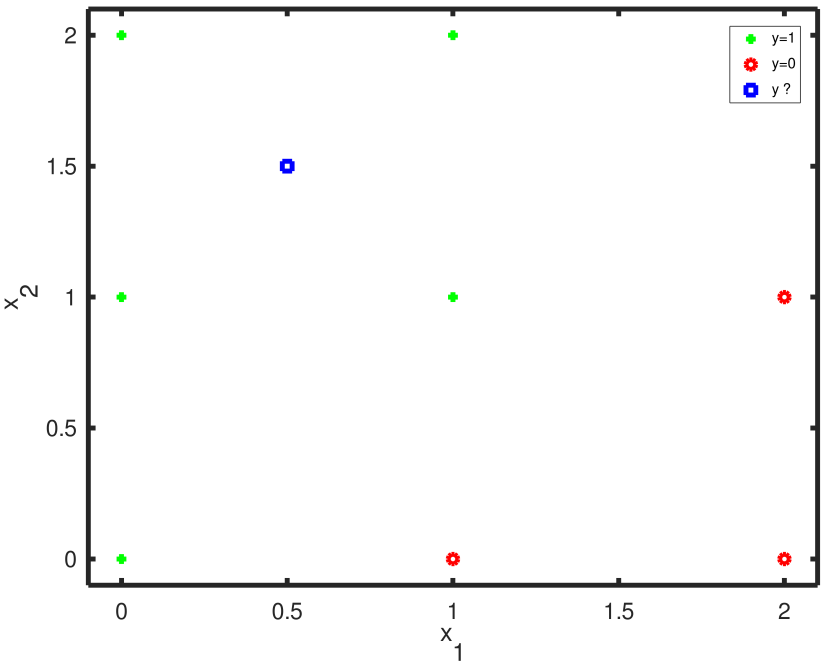
$$T\mathbf{x} = \begin{bmatrix} \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_1 \end{bmatrix} \text{ and } TY = \begin{bmatrix} y_2 \\ y_3 \\ y_1 \end{bmatrix}$$

There is a one-to-one relation between $(\mathbf{x}, Y)$ and $(T\mathbf{x}, TY)$.

How do we know if $(T\mathbf{x})^T$ is $[\mathbf{x}_2^T, \mathbf{x}_3^T, \mathbf{x}_1^T]$ or $[\mathbf{x}_3^T, \mathbf{x}_1^T, \mathbf{x}_2^T]$

$$T \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0\times1 + \mathbf{1}\times2 + 0\times3 \\ 0\times1 + 0\times2 + \mathbf{1}\times3 \\ 1\times1 + 0\times2 + 0\times3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

What is classification?



- Query sample: blue square
- Given the training set, is it more likely to be green ($y = 1$) or red ($y = 0$)?

Conclusion of section 1, Classification of hyperspectral images

- The classification of hyperspectral images yields a classification map and hence an interpretation.
- Need of ground truth data to learn information
- Need of some belief
- Numerical complexity is an issue, here out of the scope of this lecture
- Choice of a technique should take into account what the technique is meant for.

What are classifiers

In the next section, we discuss of two simple classifiers.

Table of Contents

Content of section 2, Learning regarded as an optimization Problem

Predictor function

- Predictor function

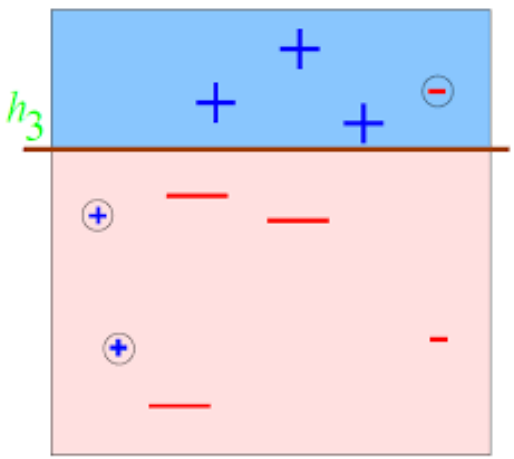
$$\widehat{y} = f(\mathbf{x})$$

- Iverson bracket

$$\delta(\Pi) = \begin{cases} 1 \text{ if } \Pi \text{ is true} \\ 0 \text{ if not} \end{cases}$$

- Sample $\mathbf{x}$ is row-vector.
- y is the label 0 or 1.

Decision stumps: example



Exercise 4

We are considering the following predictor which is an example of decision stump.

$$f_{a,b}(x) = (2a-1)\delta(x \leq b) + 1 - a$$

with a and b as parameters.

1. *Compute $f_{1,2}(0.5)$, $f_{1,0.5}(2)$.*
2. *Prove that*

$$f_{x,y}(z) = f_{x,z}(y)\delta(y = z) + (1 - f_{x,z}(y))\delta(y \neq z)$$

Integers representing binaries

Logic	$-1, +1$	$0, 1$
$y =$ POSITIVE	$y = +1$	$y = 1$
$y =$ NEGATIVE	$y = -1$	$y = 0$
$y_1 = y_2$	$y_1 y_2$	$\delta(y_1 = y_2)$ $= y_1 y_2 + (1 - y_1)(1 - y_2)$ $= (2y_1 - 1)y_2 + (1 - y_1)$ $= 0.5\widetilde{y}_1\widetilde{y}_2 + 0.5$

$$\widetilde{y} = 2y - 1 \text{ and } y = 0.5\widetilde{y} + 0.5$$

Answer to exercise 4

$$f_{a,b}(x) = (2a-1)\delta(x \le b) + 1 - a$$

$$f_{x,y}(z) = f_{x,z}(y)\delta(y = z) + (1 - f_{x,z}(y))\delta(y \neq z)$$

1. $f_{1,2}(0.5) = (2\times 1 - 1)\delta(0.5 \le 2) + 1 - 1 = 1$
   $f_{1,0.5}(2) = (2\times 1 - 1)\delta(2 \le 0.5) + 1 - 1 = 0$
2. Assuming $y = z$, $f_{x,y}(z) = f_{x,z}(z) = f_{x,z}(y)$

   Assuming $y \neq z$, $f_{x,y}(z) = (2x-1)\delta(z \le y) + 1 - x$
   $= (2x-1)(1 - \delta(y < z)) + 1 - x = (1-2x)\delta(y \le z) + x$
   $= -(2x-1)\delta(y \le z) + 1 - (1-x) = 1 - f_{x,z}(y)$

Decision stumps: definition

A **decision stump** makes a decision based on the value of a feature.

$$f_{\theta_F,\theta_x,\theta_y}(\mathbf{x}) = (2\theta_y - 1)\delta(x_{\theta_F} \leq \theta_x) + 1 - \theta_y \quad (1)$$

with $\theta_y \in \{0, 1\}$, $\theta_F \in \{1 \dots F\}$ and $\theta_x \in \mathbb{R}$

Scalar product

The **feature space** is the set comprising all possible values of $\mathbf{x}$. We define on it a scalar product

$$\mathbf{x}.\mathbf{x}' = \sum_{f=1}^{F} x_f x'_f$$

This scalar product can be written with matrix operations.

$$\mathbf{x}.\mathbf{x}' = \mathbf{x}\mathbf{x}'^T$$

Note that the transpose operation would apply on the first element if $\mathbf{x}$ and $\mathbf{x}'$ were column vectors.

Example of linear predictor

Exercise 5

We consider a predictor f defined as

$$f(\mathbf{x}) = \delta(2x_1 + x_2 \leq 2) \tag{2}$$

1. *Rewrite f using the scalar product.*
2. *Rewrite f using matrix operations.*
3. *Plot $x_1 \mapsto f([x_1, 0])$.*
4. *Plot $x_2 \mapsto f([0, x_2])$.*

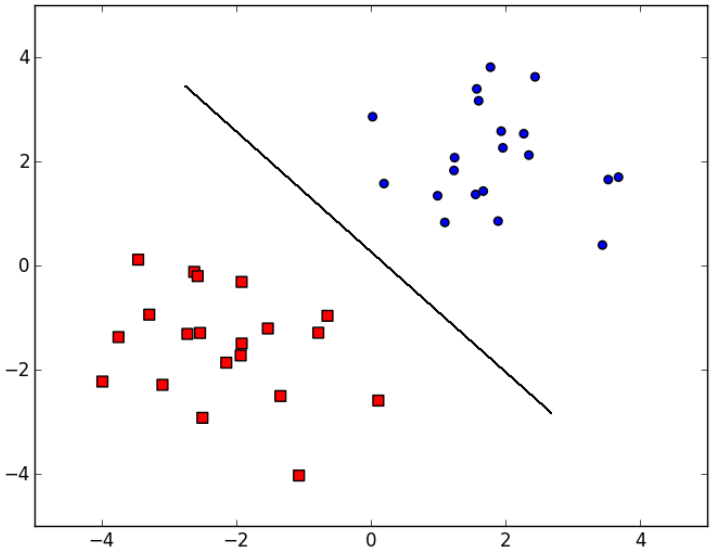
We are considering two sets

$$\mathscr{X}_0 = \{\mathbf{x} \,|\, f(\mathbf{x}) = 0\} \text{ and } \mathscr{X}_1 = \{\mathbf{x} \,|\, f(\mathbf{x}) = 1\}$$

6. *Plot the line separating the two sets and indicate which set is where?*

Linear predictors

$$f_{\mathbf{a},b}(\mathbf{x}) = \delta(\mathbf{a}.\mathbf{x} \leq b)$$



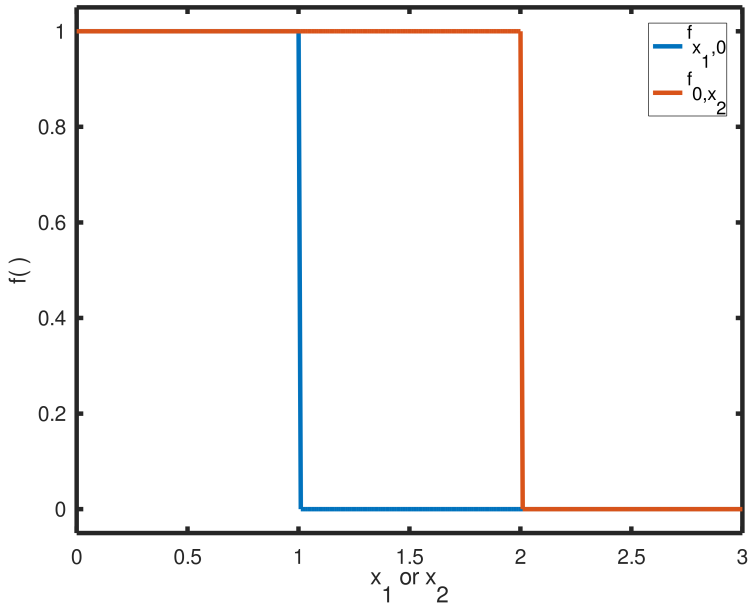
Remark

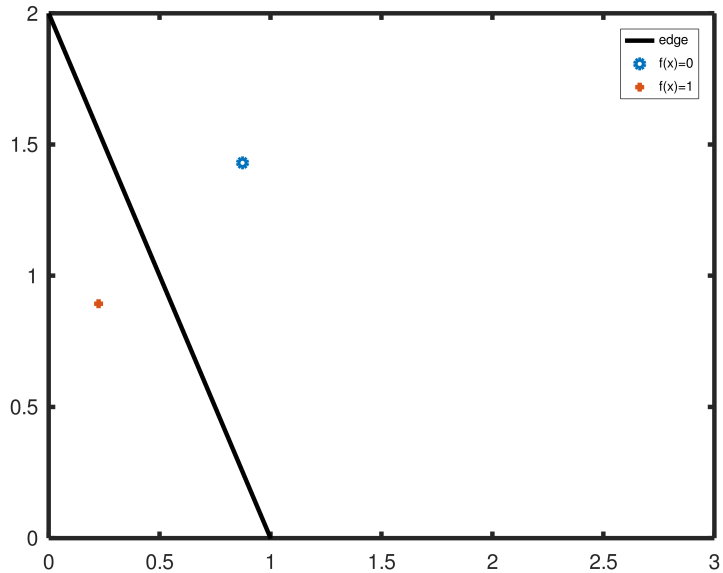
When $b > 0$, for any $\lambda > 0$, $f_{\mathbf{a},b}(\mathbf{x}) = f_{\lambda\mathbf{a},\lambda b}(\mathbf{x})$ This property shows that the proposed model is not **non-identifiable**. Note that if we use only **a** to define this predictor, then we need some extra information.

Answer to exercise 5

$$f(\mathbf{x}) = \delta(2x_1 + x_2 \leq 2)$$

1. Let $\mathbf{u} = [1\,2]$,
   $f(\mathbf{x}) = \delta(\mathbf{x}.\mathbf{u} \leq 2)$.
2. $f(\mathbf{x}) = \delta(\mathbf{x}\mathbf{u}^T \leq 2)$.
3. $f([x_1, 0]) = \delta(x_1 \leq 1)$
4. $f([0, x_2]) = \delta(x_2 \leq 2)$
5. Let $x_2 = g(x_1)$ be the edge.
   $g(x_1) = 2 - 2x_1$.





Conclusion of subsection 1, Decision stump and linear classifier

- Binary context: 2 classes
- Decision stumps and linear classifiers are predictor functions
- They act on the feature space
- They are defined by a parameter here $\theta_F, \theta_x, \theta_y$ or $b, \mathbf{a}$
- Given a query sample $\mathbf{x}$, they give a prediction $\widehat{y}$

How can we compute the parameters defining the predictor functions?

In the next subsection, we discuss metrics designed for assessing predictor functions.

Content of section 2, Learning regarded as an optimization Problem

Accuracy vs loss functions

Accuracy (overall accuracy)

what is at stake?

$$\mathcal{A}(Y, \widehat{Y}) = \frac{1}{N} \sum_{i=1}^{N} \delta(\widehat{y}_n = y_n)$$

Example of loss function

$$\mathcal{L}(Y, \widehat{Y}) = -\mathcal{A}(Y, \widehat{Y})$$

Notations

Y and $\widehat{Y}$ are column vectors stacking y_n and $\widehat{y}_n$. y_n is the true label and $\widehat{y}_n$ is the label predicted using $\mathbf{x}_n$.

Note that in $\mathcal{L}(Y, \widehat{Y})$, depending on the choice $\widehat{y}_n$ could be a real number and not a boolean in $\{0, 1\}$. This is up to the choice of the technique. Now it is not depending on $\mathbf{x}$.

Exercise 6

We are considering the predictor $f_{a,b}(x)$ *defined as*

$$f_{a,b}(x) = (2a-1)\delta(x \leq b) + 1 - a$$

with a *and* b *as parameters. and the following database* $\mathscr{S}_1$

$$\begin{array}{ll} x_1 = 1 & y_1 = 1 \\ x_2 = 1.5 & y_2 = 0 \\ x_3 = 6 & y_3 = 1 \\ x_4 = 3 & y_4 = 1 \\ x_5 = 0.5 & y_5 = 0 \end{array}$$

1. *Plot the function defined by* $b \mapsto \mathcal{A}(\mathscr{S}_1, f_{1,b})$.
2. *Plot the function defined by* $b \mapsto \mathcal{A}(\mathscr{S}_1, f_{0,b})$.
3. *Select values for* a *and* b *maximizing* $\mathcal{A}(\mathscr{S}_1, f_{a,b})$.
4. *Find the corresponding maximum value of* $\mathcal{A}(\mathscr{S}_1, f_{a,b})$.
5. *Use* argmax *and* max *to write the answers to the two last questions.*

Answer to exercise 6

$$f_{a,b}(x) = (2a-1)\delta(x \leq b) + 1 - a$$

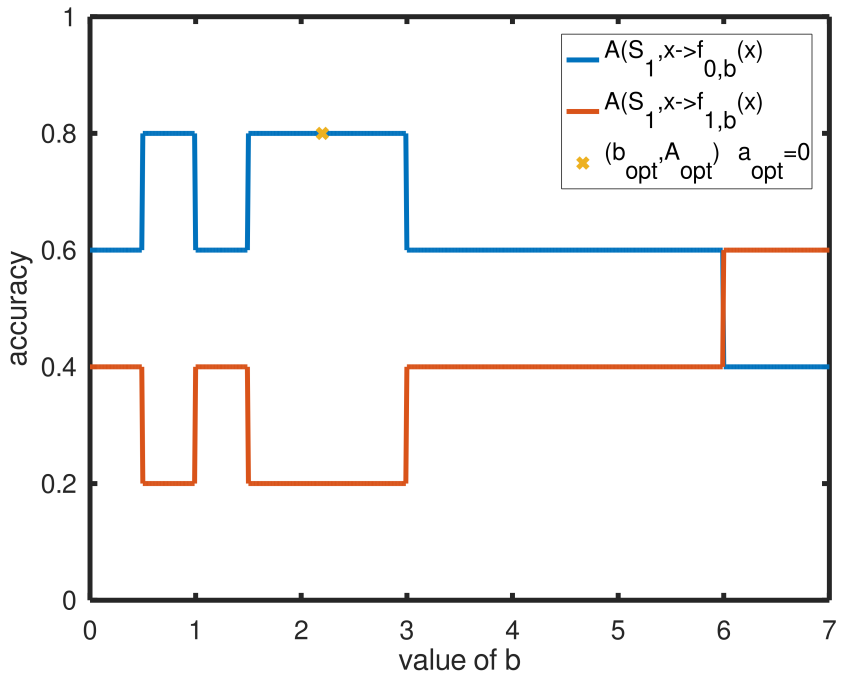
$x_1 = 1$ $\quad y_1 = 1$

$x_2 = 1.5$ $\quad y_2 = 0$

$x_3 = 6$ $\quad y_3 = 1$

$x_4 = 3$ $\quad y_4 = 1$

$x_5 = 0.5$ $\quad y_5 = 0$



1. $a = 1$

   $\delta(y_1 = \widehat{y}_{1,b}) = \delta(b \geq 1)$

   $\delta(y_2 = \widehat{y}_{2,b}) = \delta(b < 1.5)$

   $\delta(y_3 = \widehat{y}_{3,b}) = \delta(b \geq 6)$

   $\delta(y_4 = \widehat{y}_{4,b}) = \delta(b \geq 3)$

   $\delta(y_5 = \widehat{y}_{5,b}) = \delta(b < 0.5)$

2. $a = 0$

   $\delta(y_n = \widehat{y}_{n,0,b}) = 1 - \delta(y_n = \widehat{y}_{n,1,b})$

3. $a_{\text{opt}} = 0$ and $b_{\text{opt}} = 2.2$

4. $\mathcal{A}_{\text{opt}} = 0.8.$

5. $$a_{\text{opt}}, b_{\text{opt}} = \underset{a,b}{\text{argmax}} \mathcal{A}(\mathscr{S}_1, f_{a,b})$$

   $$\mathcal{A}_{\text{opt}} = \max_{a,b} \mathcal{A}(\mathscr{S}_1, f_{a,b})$$

Conclusion of subsection 2, Accuracy and loss functions

- Accuracy and loss functions tell us whether a predictor function is consistent with a dataset.
- $\mathcal{A}$ is the accuracy. It is expected to be the more appropriate metric (this depends on the application).
- Loss functions denoted $\mathcal{L}$ are less appropriate. We will see examples.
- Here higher values of $\mathcal{A}$ and lower values of $\mathcal{L}$ indicate better performance.
- In the binary context $\widetilde{y} \in \{-1, 1\}$ can be more appropriate than $y \in \{0, 1\}$.

How these metrics are going to help us finding the parameters. $\theta_F, \theta_x, \theta_y$ or $b, \mathbf{a}$.

Parameters are chosen with respect to these metrics.

Content of section 2, Learning regarded as an optimization Problem

Optimization problem

The loss function is a proxy indicating how to approach the goal.

- Parameters are selected so that
$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} \mathcal{L}(Y, [f^v_\Theta(\mathbf{x}_n)]_n)$$
where $f^v_\Theta(\mathbf{x})$ is a real-valued function.
- Real-valued predictor
$$f^v(\mathbf{x}) \in \mathbb{R}$$
(the dependency w.r. to Θ is often omitted for the sake of clarity)
- Linear real-valued predictor
$$f^v(\mathbf{x}) = b - \mathbf{a}.\mathbf{x}$$
- L2-loss function
$$\mathcal{L}(\mathscr{S}, f^v) = \frac{1}{2}\sum_{n=1}^{N}(f^v(\mathbf{x}_n) - \widetilde{y}_n)^2$$

Exercise 7

We are considering the following 2-feature data set denoted $\mathscr{S}_2$.

$$\begin{array}{lll} x_{11} = 2 & x_{12} = 0.5 & y_1 = 1 \\ x_{21} = 1 & x_{22} = 2 & y_2 = 0 \\ x_{31} = 0 & x_{32} = 0 & y_3 = 1 \end{array}$$

We consider a family of predictors $f_{\mathbf{a},b}$ *defined as*

$$f_{\mathbf{a},b}(\mathbf{x}) = \delta(\mathbf{a}.\mathbf{x} \leq b)$$

with $\mathbf{a} = [a_1, a_2]$.
We define $\mathscr{J}(a_1, a_2, b) = \mathscr{L}(\mathscr{S}_2, f_{\mathbf{a},b})$

1. *Compute* $\mathscr{J}(a_1, a_2, b)$ *as the sum of three quadratic expressions. And explain why* 0 *an obvious lower bound of* $\mathscr{J}$ *is likely to be reached.*
2. *Show that* $\mathscr{J}(a_1, a_2, b) = 0$ *if this system is solved.*

$$\begin{cases} 2a_1 + 0.5a_2 - b = -1 \\ a_1 + 2a_2 - b = 1 \\ b = 1 \end{cases}$$

3. *Solve the system and show that* $a_1 = -\frac{2}{7}$, $a_2 = \frac{8}{7}$ *and* $b = 1$.

Answer to exercise 7 I

1. $$\mathcal{J}(b, \mathbf{a}) = \mathcal{L}(\mathscr{S}, f^v) = \frac{1}{2}\sum_{n=1}^{3}(b - \mathbf{a.x} - \widetilde{y}_n)^2$$
   Square values are necessarily non-negative so $\mathcal{J}(b, \mathbf{a}) \geq 0$. This lower bound is the actual minimum value if these square values are zeroed, that is if three constrained equations are met by three free variables b, a_1, a_2.

2. $$2\mathcal{J}(b, \mathbf{a}) = (b - 2a_1 - 0.5a_2 - 1)^2 + (b - a_1 - 2a_2 + 1)^2 + (b - 1)^2$$
   $\mathcal{J}(b, \mathbf{a}) = 0$ iff
   $$\begin{cases} 2a_1 + 0.5 * a_2 - b = -1 \\ a_1 + 2a_2 - b = 1 \\ b = 1 \end{cases}$$

Answer to exercise 7 II

3. $$\mathcal{J}(1,[-2/7,8/7]) = (1 - 2 * (-2/7) - 0.5 * 8/7 - 1)^2$$
$$+(1 - (-2/7) - 2(8/7) + 1)^2 + (1 - 1)^2 = 0$$

Need of a more general technique

In the example shown in exercise 7, we have three samples and three free variables

$$\min_{\mathbf{a},b} \mathcal{J}(\mathbf{a}, b) = 0 \text{ and } \mathbf{a}, b = \underset{\mathbf{a},b}{\operatorname{argmin}} \mathcal{J}(\mathbf{a}, b)$$

In general this is not true.

- Finding a solution using an algorithm
- Using linear algebra.

Conclusion of subsection 3, Optimization problem

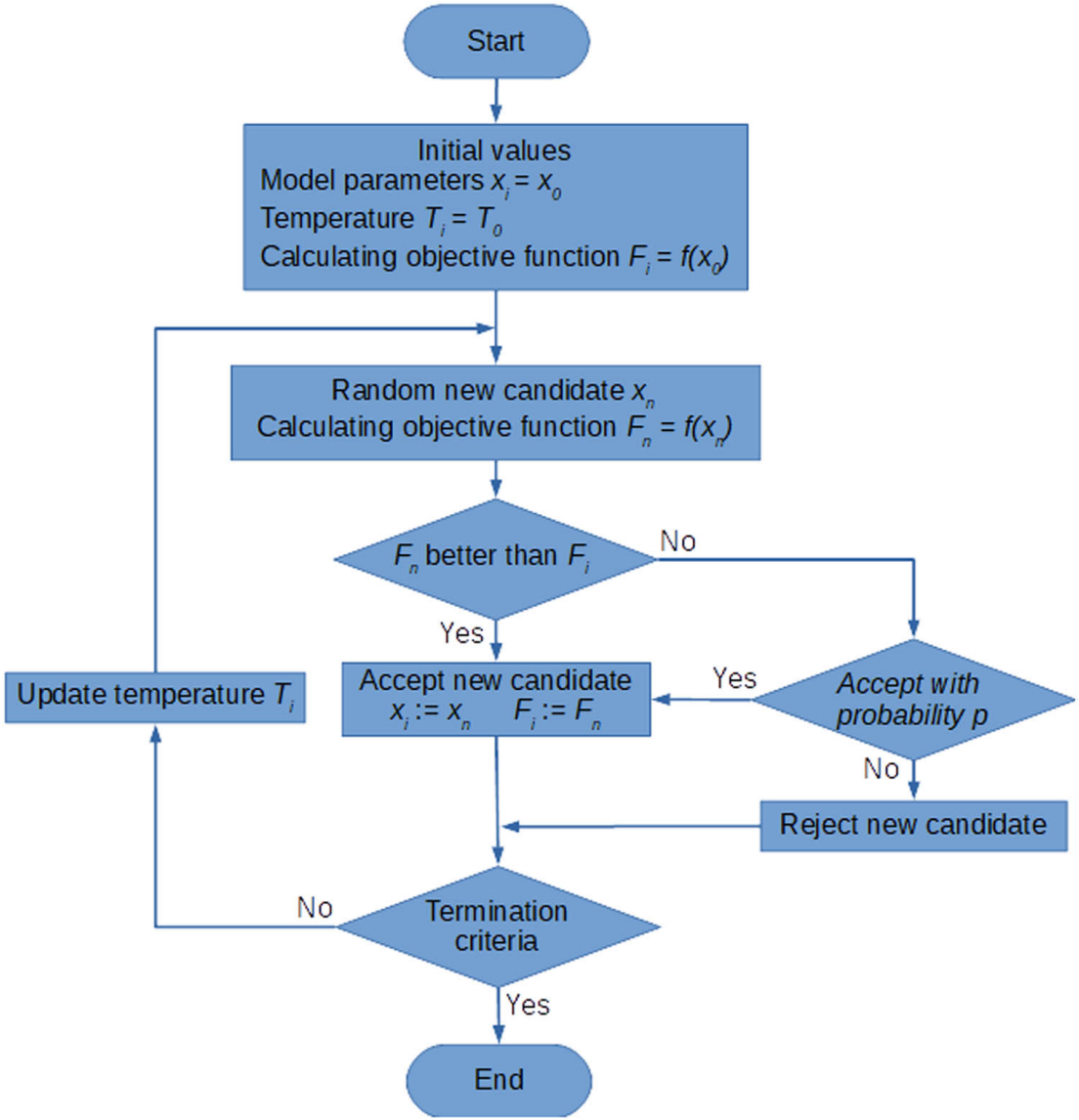
- Parameters of a predictor function are chosen so as to minimize or maximize a loss function or the accuracy for a given dataset.
- An $L2$-loss function is an example.
- It works like a regression, as if we wanted to predict a real value for $\widetilde{y}$.

Even a simple example seems to require complex computations, how are we going to deal with more complex examples?

In the next section, we will see an example of algorithm. And in the section after, we will see how we can make use of linear algebra and matrices.

Content of section 2, Learning regarded as an optimization Problem

Simulated annealing (a more complex kind)



Simplified simulated annealing

Require: Loss function
Ensure: Θ parameters minimizing the loss function.

1: Select randomly Θ and set $L := +\infty$.
2: **for** k=1:10000 **do**
3: Select randomly r, a real in $[0, 6]$ and set $\sigma := 10^{-r}$.
4: Select randomly $\Delta\Theta$ along a centered Gaussian distribution with σ as standard deviation.
5: **if** $\mathcal{L}(\Theta + \Delta\Theta) < L$ **then**
6: Set $\Theta := \Theta + \Delta\Theta$ and $L := \mathcal{L}(\Theta)$.
7: Display Θ.

Using simulated_annealing.m

```
cost_function=@(theta)(theta(1)-2)^2+(theta(2)-3)^2;
dim=2;
theta=simulated_annealing(cost_function,dim);
```

The code displays

```
L=28.2762
L=25.1406
L=23.7017
L=15.3473
```

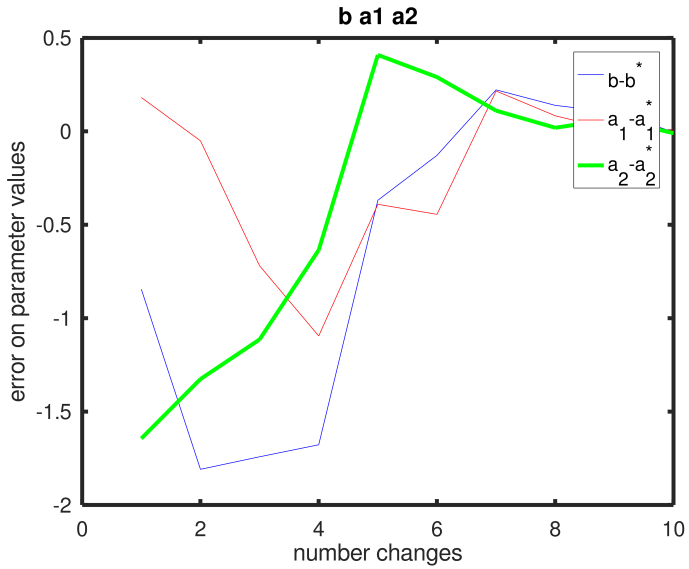
We have the best parameter found with

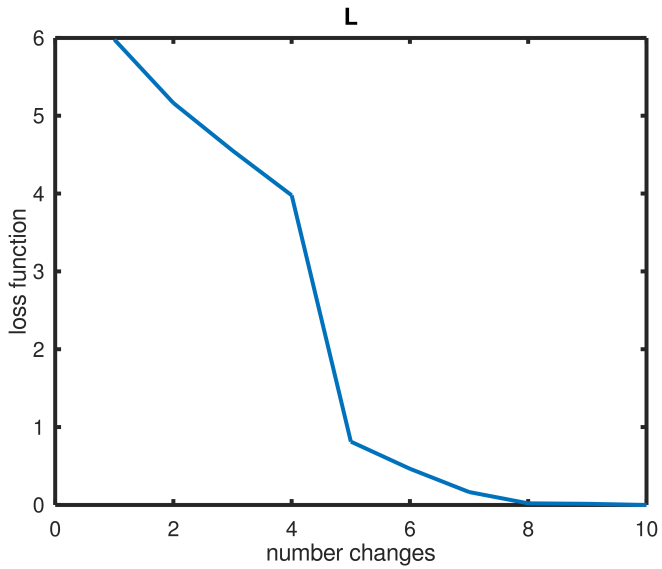
```
octave:24> theta
theta =

   1.9994
   3.0029
```

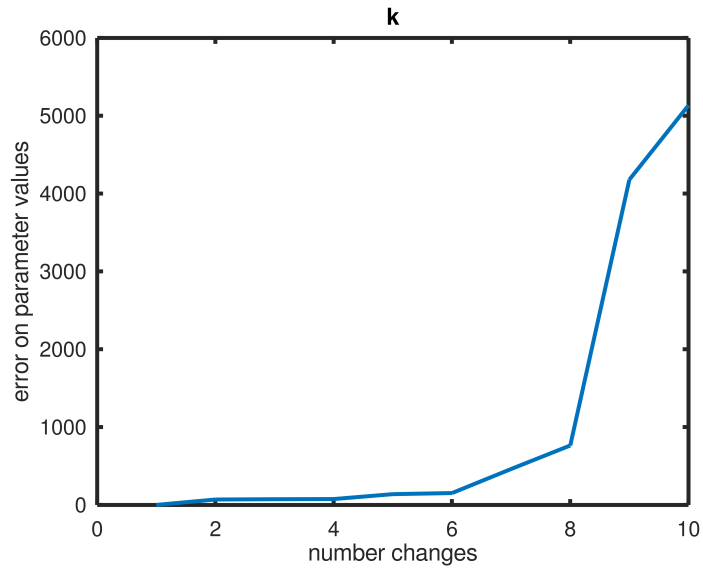
Exercise 8

Give the Octave code that uses `simulated_annealing` *to find an approximation of* $\mathbf{a}$ *and* a *of exercise 7.*

Example of poor performances with simulated annealing







Answer to exercise 8 I

```
function J=J2(theta)
  x1=[2 0.5]; y1=1;
  x2=[1 2]; y2=0;
  x3=[0 0]; y3=1;
  tilde=@(y)2*y-1;
  b=theta(1); a1=theta(2); a2=theta(3);
  J=(b-a1*x1(1)-a2*x1(2)-tilde(y1))^2;
  J=J+(b-a1*x2(1)-a2*x2(2)-tilde(y2))^2;
  J=J+(b-a1*x3(1)-a2*x3(2)-tilde(y3))^2;
end
theta=simulated_annealing(@(theta)J2(theta),3);
```

Conclusion of subsection 4, Simulated annealing

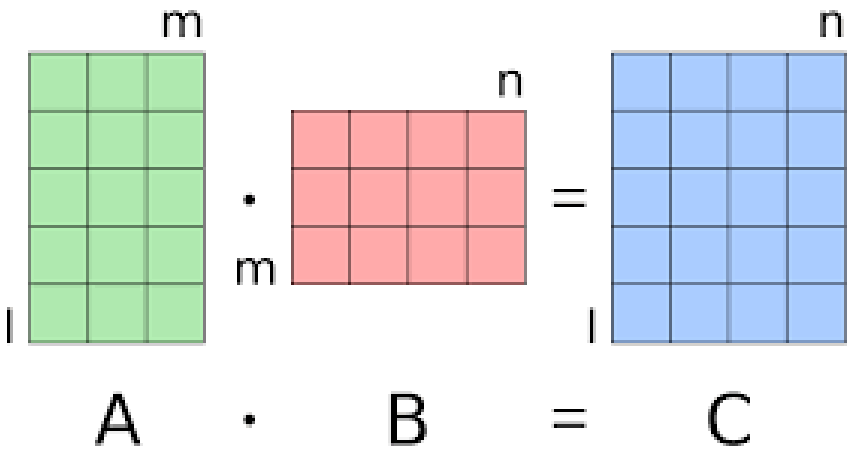
- Simulated annealing is quicker than a uniform random search.
- It refines the search after some iterations.
- The choice of the proposed algorithm is to make it easy to use at the expense of a high numerical complexity.

An other technique to select parameters with respect to a loss function and a dataset?

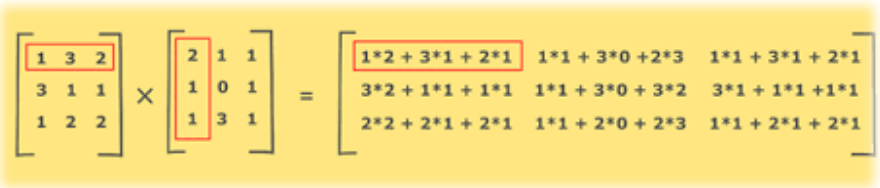
In the next subsection, we discuss the minimization of the $L2$-loss function for linear classifiers.

Content of section 2, Learning regarded as an optimization Problem

Product of two matrices



$$C = AB$$



$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}$$

biais considered as a supplementary feature

Predicting function

$$f^v(\mathbf{x}) = b - \mathbf{a}.\mathbf{x} = \mathbf{w}.\overset{\Delta}{\mathbf{x}}$$

We use the following definition

$$\mathbf{w} = [-a_1 \; -a_2 \; \ldots \; -a_F \; b] = [-\mathbf{a} \; b]$$

$$\overset{\Delta}{\mathbf{x}} = [x_1 \; x_2 \; \ldots \; x_F \; 1] = [\mathbf{x} \; 1]$$

The matrix definition of $\mathbf{X}$ is modified into

$$\overset{\Delta}{\mathbf{X}} = \begin{bmatrix} \mathbf{x}_1 & 1 \\ \vdots & 1 \\ \mathbf{x}_N & 1 \end{bmatrix} = \begin{bmatrix} \overset{\Delta}{\mathbf{x}}_1 \\ \vdots \\ \overset{\Delta}{\mathbf{x}}_N \end{bmatrix} = \begin{bmatrix} & 1 \\ \mathbf{X} & \vdots \\ & 1 \end{bmatrix}$$

Expressing the loss function with matrices I

$$[\ldots\ldots\ldots] \begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

Scalar product as vector multiplication

$$\mathbf{w}.\overset{\Delta}{\mathbf{x}} = \mathbf{w}\overset{\Delta}{\mathbf{x}}^T$$

$$\mathcal{L}(\mathscr{S}, f^v) = \frac{1}{2}\sum_{n=1}^{N}(\mathbf{w}\overset{\Delta}{\mathbf{x}}_n^T - \widetilde{y}_n)^2$$

Expressing the loss function with matrices II

Sum of square values as vector multiplication

$$\sum_{n=1}^{N} \widetilde{y}_n = \widetilde{Y}^T \widetilde{Y}$$

In the same way,

$$\mathcal{L}(\mathscr{S}, f^v) = \frac{1}{2} \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T - \widetilde{Y} \right)^T \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T - \widetilde{Y} \right)$$

Expanding follows classical rules

$$2\mathcal{L}(\mathscr{S}, f^v) = \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T \right)^T \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T \right) - \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T \right)^T \widetilde{Y} - \widetilde{Y}^T \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T \right) + \widetilde{Y}^T \widetilde{Y}$$

Transpose of the product of two matrices

Rule

$(AB)^T = B^T A^T$ and if AB is a scalar $AB = (AB)^T = B^T A^T$
we also have $(AB)C = A(BC)$

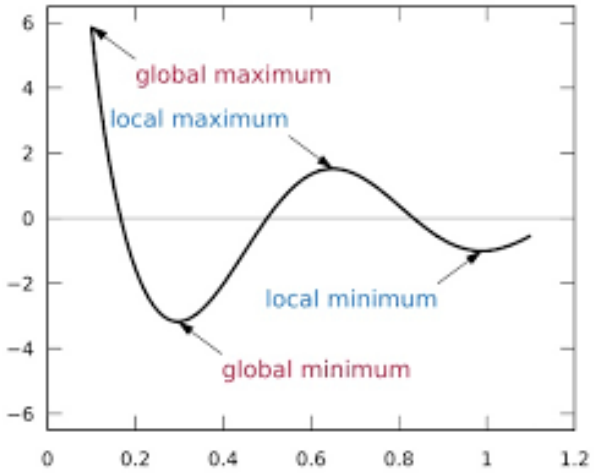
$$2\mathcal{L}(\mathscr{S}, f^v) = \left(\overset{\Delta}{\mathbf{X}}\mathbf{w}^T\right)^T \left(\overset{\Delta}{\mathbf{X}}\mathbf{w}^T\right) - \left(\overset{\Delta}{\mathbf{X}}\mathbf{w}^T\right)^T \widetilde{Y} - \widetilde{Y}^T \left(\overset{\Delta}{\mathbf{X}}\mathbf{w}^T\right) + \widetilde{Y}^T \widetilde{Y}$$

becomes

$$2\mathcal{L}(\mathscr{S}, f^v) = \mathbf{w}\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\mathbf{w}^T - 2\mathbf{w}\overset{\Delta}{\mathbf{X}}^T \widetilde{Y} + \widetilde{Y}^T \widetilde{Y}$$

We are now considering $\mathcal{J}(\mathbf{w}) = \mathcal{L}(\mathscr{S}, f^v)$

Finding a local minimum



- $\mathbf{w}_0$ is local minimum iff for all $\mathbf{w}$ in a neighborhood of $\mathbf{w}_0$, $\mathcal{J}(\mathbf{w}_0) \leq \mathcal{J}(\mathbf{w})$
- **If** $\mathbf{w}$ is a local minimum then
$$\frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}} = 0$$
- $\mathbf{w}^*$ is a global minimum iff $\forall \mathbf{w},\ \mathcal{J}(\mathbf{w}^*) \leq \mathcal{J}(\mathbf{w})$
- Under some more **involved conditions**, a unique local minimum that bounds from below all values at the domain's edges is a global minimum.

Partial derivative: definition

Rule

The derivative of a **scalar** function with respect to a **row** or a **column** vector is a **column** or a **row** vector.

$$\frac{\partial \mathcal{J}(\mathbf{w})}{\partial [w_1, w_2, \ldots, w_{F+1}]} = \begin{bmatrix} \frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}_1} \\ \frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}_2} \\ \vdots \\ \frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}_{F+1}} \end{bmatrix}$$

$$\frac{\partial \mathcal{J}(\mathbf{w})}{\partial \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_{F+1} \end{bmatrix}} = \begin{bmatrix} \frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}_1}, \cdots \frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}_{F+1}} \end{bmatrix}$$

Partial derivative: formulas

Notations

w is a **row vector** and V is a **column vector**. $\mathbf{w}V$ is a scalar and $\mathbf{w}V = V^T\mathbf{w}^T$. A is a square matrix.

if A is symmetric:
$A^T = A$

$$\frac{\partial \mathbf{w}V}{\partial \mathbf{w}} = \frac{\partial V^T\mathbf{w}^T}{\partial \mathbf{w}} = V$$
$$\frac{\partial \mathbf{w}V}{\partial \mathbf{w}^T} = \frac{\partial V^T\mathbf{w}^T}{\partial \mathbf{w}^T} = V^T$$

$$\frac{\partial \mathbf{w}A\mathbf{w}^T}{\partial \mathbf{w}} = A\mathbf{w}^T + A^T\mathbf{w}^T = (A + A^T)\mathbf{w}^T \qquad = 2A\mathbf{w}^T$$
$$\frac{\partial \mathbf{w}A\mathbf{w}^T}{\partial \mathbf{w}^T} = \mathbf{w}A + \mathbf{w}A^T = \mathbf{w}(A + A^T) \qquad = 2\mathbf{w}A$$

Derivative of $\mathcal{J}$

Cost function

$$2\mathcal{J}(\mathbf{w}) = \mathbf{w}\overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}}\mathbf{w}^T - 2\mathbf{w}\overset{\Delta}{\mathbf{X}}^T\widetilde{Y} + \widetilde{Y}^T\widetilde{Y}$$

Applying the rules and because $\overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}}$ is symmetric $((\overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}})^T = \overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}})$

$$\frac{\partial\mathcal{J}(\mathbf{w})}{\partial\mathbf{w}} = \overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}}\mathbf{w}^T - \overset{\Delta}{\mathbf{X}}^T\widetilde{Y}$$

Cancellation of the derivative

$$\mathbf{w}^T = \left(\overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}}\right)^{-1}\overset{\Delta}{\mathbf{X}}^T\widetilde{Y}$$

Instead of an optimization algorithm, we need to **inverse** a matrix (or solve a linear system).

Solving exercise 7 I

Exercise 9

We consider once again exercise 7 to solve without using the trick of zeroing $\mathcal{J}$ which usually does not work.

$$\begin{array}{lll} x_{11} = 2 & x_{12} = 0.5 & y_1 = 1 \\ x_{21} = 1 & x_{22} = 2 & y_2 = 0 \\ x_{31} = 0 & x_{32} = 0 & y_3 = 1 \end{array}$$

We consider a linear family of predictors $f_{\mathbf{a},b}$ defined as

$$f_{\mathbf{a},b}(\mathbf{x}) = \delta(\mathbf{a}.\mathbf{x} \leq b)$$

with $\mathbf{a} = [a_1, a_2]$. We consider an L2-loss function
$\mathcal{J}(a_1, a_2, b) = \mathcal{L}(\mathscr{S}_2, f_{\mathbf{a},b}) = \frac{1}{2}\sum_{n=1}^{N}(f^v(\mathbf{x}_n) - \widetilde{y}_n)^2$

1. *Define $\mathbf{w}$ with respect to $\mathbf{a}$ and b and $\overset{\Delta}{\mathbf{x}}$ with respect to x_1 and x_2.*
2. *Compute $\mathbf{X}$, $\overset{\Delta}{\mathbf{X}}$ and $\overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}}$.*

Solving exercise 7 II

Exercise

3. *Compute* Y, $\widetilde{Y}$ *and* $\overset{\Delta}{\mathbf{X}}^T \widetilde{Y}$
4. *Show that when* $a_1 = -\frac{2}{7}$, $a_2 = \frac{8}{7}$ *and* $b = 1$, *we have indeed that* $\frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}} = 0$.
5. *Let us suppose that we have an extra sample in* $\mathscr{S}_2$. *What are the sizes of the different vectors and matrices involved here.*
6. *Assuming that* $\mathbf{w}^*$ *that cancels the* $\mathcal{J}$*-derivative is a global minimum, show that*

$$\min_{\mathbf{w}} \mathcal{J}(\mathbf{w}) = \widetilde{Y}^T \widetilde{Y} - \widetilde{Y}^T \overset{\Delta}{\mathbf{X}} \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} \right)^{-1} \overset{\Delta}{\mathbf{X}}^T \widetilde{Y}$$

Answer to exercise 9 I

1. $\mathbf{w} = [-a_1, -a_2, b]$ and $\overset{\Delta}{\mathbf{x}} = [x_1, x_2, 1]$ because
$$f^v(\mathbf{w}) = b - \mathbf{a.x} = \mathbf{w}.\overset{\Delta}{\mathbf{x}}$$

2. $$\mathbf{X} = \begin{bmatrix} 2 & 0.5 \\ 1 & 2 \\ 0 & 0 \end{bmatrix} \text{ and } \overset{\Delta}{\mathbf{X}} = \begin{bmatrix} 2 & 0.5 & 1 \\ 1 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$
$$\overset{\Delta}{\mathbf{X}}^T = \begin{bmatrix} 2 & 1 & 0 \\ 0.5 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ and } \overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} = \begin{bmatrix} \mathbf{5} & 3 & 3 \\ 3 & \frac{17}{4} & \frac{5}{2} \\ 3 & \frac{5}{2} & 3 \end{bmatrix}$$
$$\mathbf{5} = 2 \times 2 + 1 \times 1 + 0 \times 0$$

3. $$Y = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \text{ and } \widetilde{Y} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \text{ and } \overset{\Delta}{\mathbf{X}}^T \widetilde{Y} = \begin{bmatrix} 1 \\ -\frac{3}{2} \\ 1 \end{bmatrix}$$

Answer to exercise 9 II

4. Knowing that

$$\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} = \begin{bmatrix} 5 & 3 & 3 \\ 3 & \frac{17}{4} & \frac{5}{2} \\ 3 & \frac{5}{2} & 3 \end{bmatrix}$$

and based on the solution found in exercise 7, we select $\mathbf{w}^* = [\frac{2}{7}, -\frac{8}{7}, 1]$.

$$\left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right) \mathbf{w}^{*T} = \begin{bmatrix} 1 \\ -\frac{3}{2} \\ 1 \end{bmatrix} = \overset{\Delta}{\mathbf{X}}^T \widetilde{Y}$$

5. We consider four samples.
   - The size of Y and $\widetilde{Y}$ is 4×1.
   - The size of $\mathbf{X}$ is 4×2.
   - The size of $\overset{\Delta}{\mathbf{X}}$ is 4×3.

   The remaining sizes are unchanged.
   - The size of $\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}$ and $\left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1}$ is 3×3.

Answer to exercise 9 III

- The size of $\overset{\Delta}{\mathbf{X}}^T \widetilde{Y}$ is 3×1.
- The size of $\mathbf{w}$ is 1×3.

6. We assume that $\mathbf{w}^*$ is a global minimum.

$$(\mathbf{w}^*)^T = \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1} \overset{\Delta}{\mathbf{X}}^T \widetilde{Y}$$

$$\mathbf{w}^* = \widetilde{Y}^T \overset{\Delta}{\mathbf{X}} \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1}$$

We plug this in the definition of $\mathcal{J}$.

$$\mathcal{J}(\mathbf{w}^*) = \widetilde{Y}^T \overset{\Delta}{\mathbf{X}} \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1} \overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1} \overset{\Delta}{\mathbf{X}}^T \widetilde{Y}$$

$$-2\widetilde{Y}^T \overset{\Delta}{\mathbf{X}} \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1} \overset{\Delta}{\mathbf{X}}^T \widetilde{Y} + \widetilde{Y}^T \widetilde{Y}$$

Answer to exercise 9 IV

After simplification we get the expected result.

$$\mathcal{J}(\mathbf{w}^*) = \widetilde{Y}^T\widetilde{Y} - \widetilde{Y}^T\overset{\Delta}{\mathbf{X}}\left(\overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}}\right)^{-1}\overset{\Delta}{\mathbf{X}}^T\widetilde{Y}$$

Comment on exercise 9

Remark 1

This least square technique is good for regression, not so much for classification as we will see later on.

Remark 2

Techniques that can be defined with matrices are generally easier to implement. It is easier to check the implementation.

Conclusion of subsection 5, Method of least squares

- Matrix formulas: product, transposition, expanding rules.
- Derivative of a scalar function with respect to a vector.
- First use of $\mathbf{X}^T\mathbf{X}$ also called covariance matrix.
- Definition of $\overset{\Delta}{\mathbf{X}}$.
- Parameter values are obtained by minimizing $\overset{\Delta}{\mathbf{X}}^T\overset{\Delta}{\mathbf{X}}$.

These are techniques requiring the knowledge of Y

In the next section we discuss technique not needing Y.

Content of section 2, Learning regarded as an optimization Problem

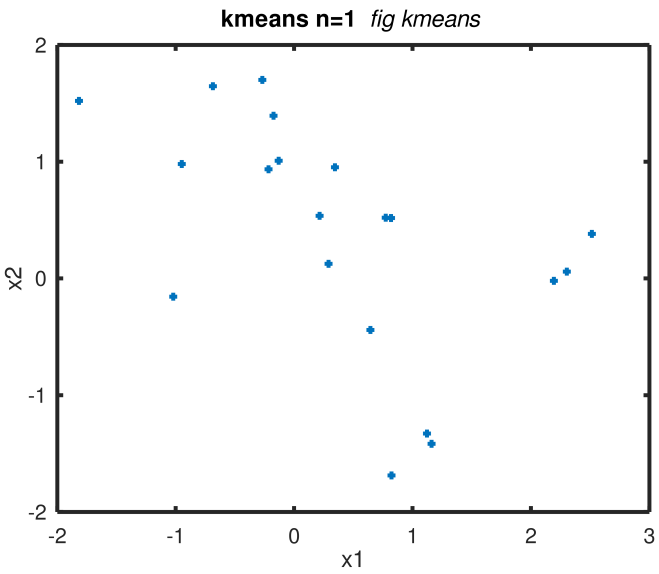
Unsupervised classification

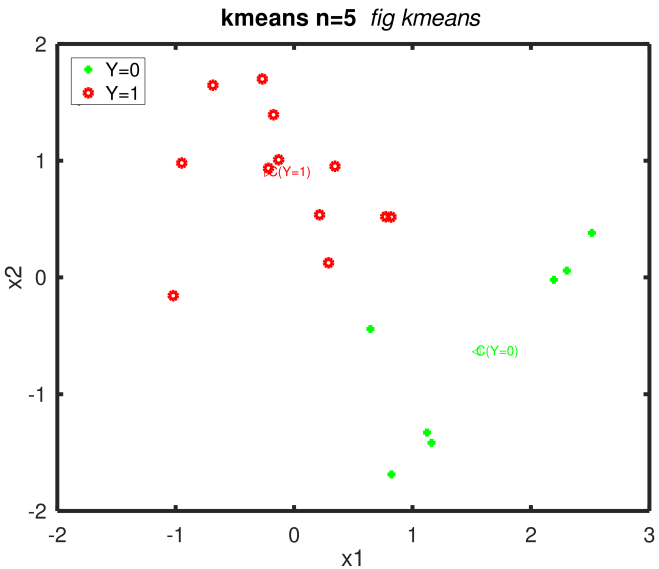
Definition

Of the dataset $(\mathbf{X}, Y)$, only $\mathbf{x}$ is used. $(\mathbf{X}^T = [\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T])$

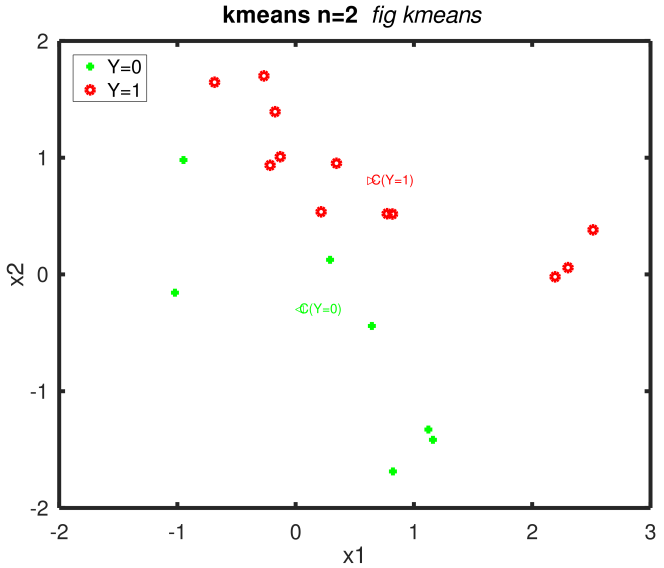
Clusters

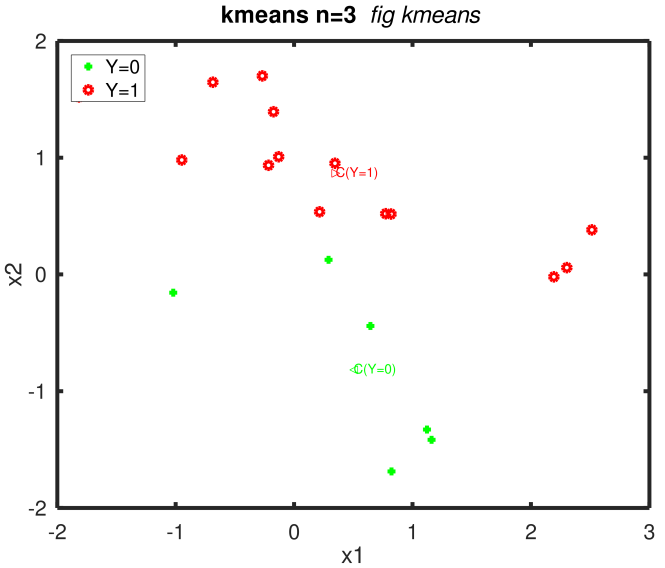
Instead of classes, we consider clusters.

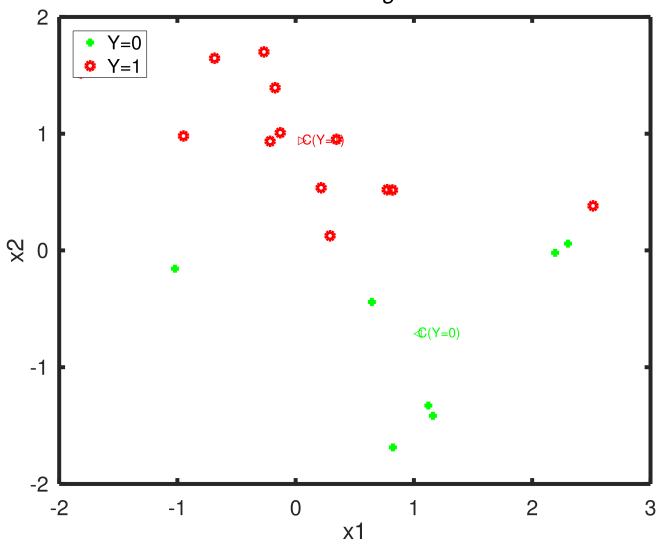


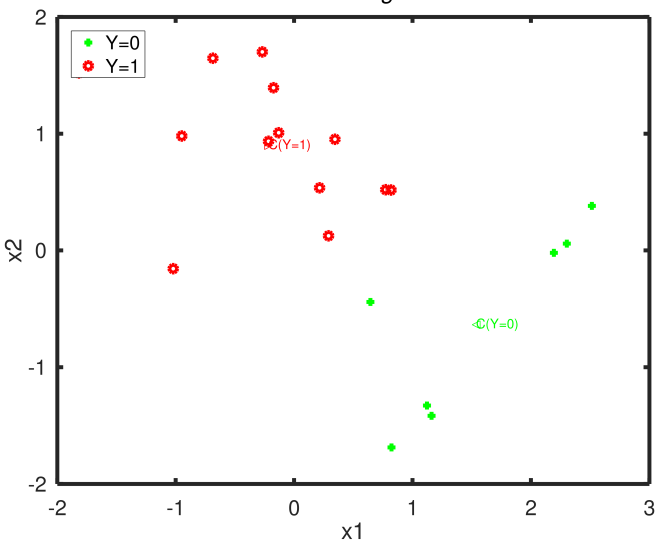


kmeans


kmeans n=2 fig kmeans
Y=0
Y=1
C(Y=1)
C(Y=0)
x1
x2


kmeans n=3 fig kmeans
Y=0
Y=1
C(Y=1)
C(Y=0)
x1
x2


kmeans n=4 fig kmeans
Y=0
Y=1
C(Y=0)
x1
x2


kmeans n=5 fig kmeans
Y=0
Y=1
C(Y=1)
C(Y=0)
x1
x2

Algorithm of kmeans

Exercise 10

We consider a set of points $\mathbf{X}$ *and two clusters. Two points are first randomly selected. Then the two following iterations are repeated.*

- *Each point is assigned to the closest point.*
- *Each geometric center is updated with its new and removed members.*

1. *Give the algorithm*

Answer to exercise 10

Require: $\mathbf{X}$
Ensure: Y

1: Select randomly two rows of $\mathbf{x}$: $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$.
2: Set Y with zeros.
3: **repeat**
4: $\quad Y_{\text{old}} = Y$
5: $\quad$ **for** $n = 1 : N$ **do**
6: $\quad\quad y_n = \delta(d(\mathbf{x}_n, \boldsymbol{\mu}_0) > d(\mathbf{x}_n, \boldsymbol{\mu}_1))$
7: $\quad Y = [y_1 \dots y_N]^T$
8: $\quad \boldsymbol{\mu}_0 = \frac{1}{\#\{n|y_n=0\}} \sum_{y_n=0} \mathbf{x}_n$
9: $\quad \boldsymbol{\mu}_1 = \frac{1}{\#\{n|y_n=1\}} \sum_{y_n=1} \mathbf{x}_n$
10: **until** $Y = Y_{\text{old}}$

An ad hoc loss function

The number of samples assigned to each cluster is

$$N_0(Y) = \sum_{n=1}^{N} \delta(y_n = 0) = \sum_{n=1}^{N} 1 - y_n \text{ and } N_1(Y) = \sum_{n=1}^{N} \delta(y_n = 1) = \sum_{n=1}^{N} y_n$$

Given a set of assignments indicated with Y, we define the geometric center of the two clusters in the feature space

$$\boldsymbol{\mu}_0(\mathbf{X}, Y) = \frac{1}{N_0(Y)} \sum_{n=1}^{N} (1 - y_n)\mathbf{x}_n$$

$$\boldsymbol{\mu}_1(\mathbf{X}, Y) = \frac{1}{N_1(Y)} \sum_{n=1}^{N} y_n \mathbf{x}_n$$

We derive a **norm** from the scalar product

$$\|\mathbf{x}\|^2 = \mathbf{x}.\mathbf{x}$$

We define a modified kind of **within point scatter**

$$J(\mathbf{X}, Y) = \sum_{n=1}^{N} (1 - y_n)\|\mathbf{x}_n - \boldsymbol{\mu}_0(\mathbf{X}, Y)\|^2 + \sum_{n=1}^{N} y_n \|\mathbf{x}_n - \boldsymbol{\mu}_1(\mathbf{X}, Y)\|^2$$

This is the loss function that is non-increasing when Y is modified along kmeans.

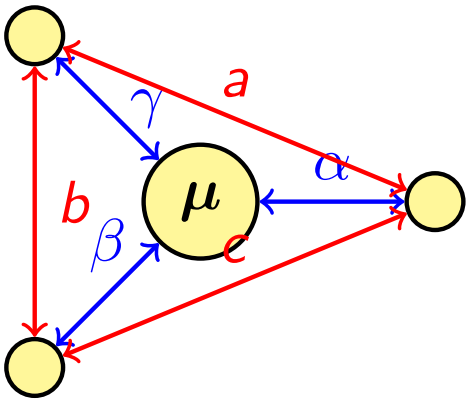
Star-triangle identity

We consider a set of N samples $\mathbf{x}_n$

$$2N\sum_{n=1}^{N}\|\mathbf{x}_n-\boldsymbol{\mu}\|^2=\sum_{n=1}^{N}\sum_{n'=1}^{N}\|\mathbf{x}_n-\mathbf{x}_{n'}\|^2$$

where the mean is given by

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

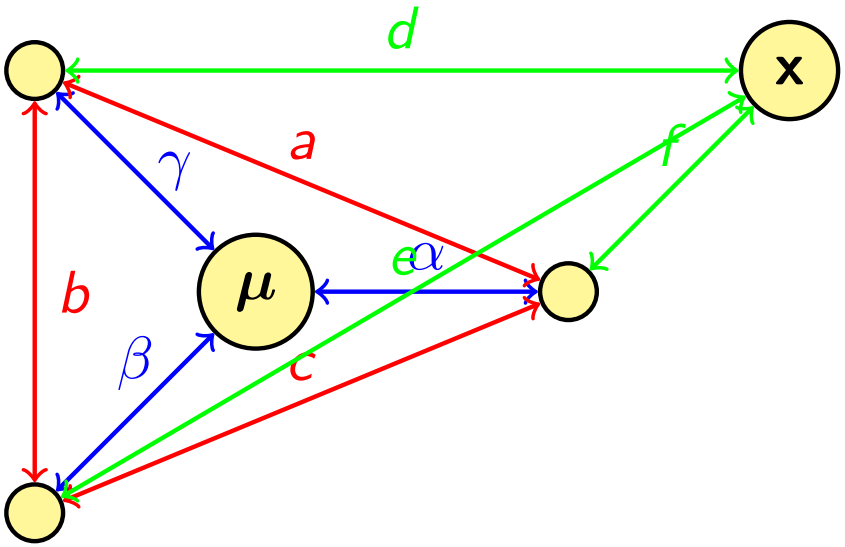


$$2\times\mathbf{3}(\alpha^2+\beta^2+\gamma^2)=2(a^2+b^2+c^2)$$

Adding-a-sample identity

We consider a set of N samples $\mathbf{x}_n$ and an extra sample $\mathbf{x}$ denoted also $\mathbf{x}_{N+1}$.

$$\sum_{n=1}^{N+1}\sum_{n=1}^{N+1}\|\mathbf{x}_n-\mathbf{x}_{n'}\|^2=\left(1+\frac{1}{N}\right)\sum_{n=1}^{N}\sum_{n=1}^{N}\|\mathbf{x}_n-\mathbf{x}_{n'}\|^2+2N\|\boldsymbol{\mu}-\mathbf{x}\|^2$$



$$2(a^2+b^2+c^2+d^2+e^2+f^2)=(1+\frac{1}{3})(a^2+b^2+c^2)+2\times3\|\boldsymbol{\mu}-\mathbf{x}\|^2$$

Optimization problem I

Exercise 11

We consider a dataset $(\mathbf{X}, Y)$ *and denote* $N_0, N_1, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1$ *the number of* 0*-labeled samples,* 1*-labeled samples, the geometric center of the* 0*-labeled samples and that of the* 1*-labeled samples.*

1. *Prove that*
$$N_0\boldsymbol{\mu}_0 + N_1\boldsymbol{\mu}_1 = N\boldsymbol{\mu} = \sum_{n=1}^{N} \mathbf{x}_n$$
where $\boldsymbol{\mu}$ *is the geometric center of the samples in the feature space.*
2. *Let* $Y' = Y$ *except for* $n = n_0$ *where* $y_{n_0} = 0$ *and* $y'_{n_0} = 1$*. Show that*
$$N_0(Y') = N_0(Y) - 1, \quad N_1(Y') = N_1(Y) + 1,$$

Optimization problem II

Exercise

3. *Let $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1$ be the means of the 0 and 1-labeled samples before the modification. Let $\boldsymbol{\mu}'_0, \boldsymbol{\mu}'_1$ be the corresponding means after the modification. Show that*
$$\boldsymbol{\mu}'_0 - \mathbf{x} = \frac{N_0}{N_0 - 1}(\boldsymbol{\mu}_0 - \mathbf{x})$$
4. *We denote by J and J' the values of loss function for $(\mathbf{X}, Y)$ and $(\mathbf{X}', Y')$. Using the adding-a-sample identity, show that*
$$J' - J = \frac{N_1}{N_1 + 1}\|\boldsymbol{\mu}_1 - \mathbf{x}\|^2 - \frac{N_0}{N_0 - 1}\|\boldsymbol{\mu}_0 - \mathbf{x}\|^2$$
5. *Show that $J' \leq J$, when Y' is modified according to kmeans, still assuming that here only* **one** *component changes.*

Answer to exercise 11 I

1. $$N_0\boldsymbol{\mu}_0 + N_1\boldsymbol{\mu}_1 = \left(\sum_{n=1}^{N}(1-y_n)\mathbf{x}_n\right) + \left(\sum_{n=1}^{N} y_n\mathbf{x}_n\right) = \sum_{n=1}^{N}\mathbf{x}_n$$
2. Observing that $y'_n = y_n + \delta(n = n_0)$, we get
   $$N_0(Y') = \sum_{n=1}^{N} 1 - y'_n = \left(\sum_{n=1}^{N} 1 - y_n\right) - 1 = N_0(Y) - 1$$
   $$N_1(Y') = \sum_{n=1}^{N} y'_n = \left(\sum_{n=1}^{N} y_n\right) + 1 = N_1(Y) + 1$$
3. When a new element is removed from the geometric-center computation, we have
   $$(N_0 - 1)\boldsymbol{\mu}'_0 = N_0(Y')\boldsymbol{\mu}'_0 = N_0(Y)\boldsymbol{\mu}_0 - \mathbf{x} = N_0\boldsymbol{\mu}_0 - \mathbf{x}$$
   We then get
   $$\begin{aligned}(N_0 - 1)(\boldsymbol{\mu}'_0 - \mathbf{x}) &= (N_0 - 1)\boldsymbol{\mu}'_0 - (N_0 - 1)\mathbf{x} \\ &= N_0\boldsymbol{\mu}_0 - \mathbf{x} - (N_0 - 1)\mathbf{x} = N_0(\boldsymbol{\mu} - \mathbf{x})\end{aligned}$$
   and hence that $\boldsymbol{\mu}'_0 - \mathbf{x} = \frac{N_0}{N_0 - 1}(\boldsymbol{\mu} - \mathbf{x})$

Answer to exercise 11 II

4. We consider two sets of samples: $\mathscr{X}_0'$ is the set of 0-labeled samples after modification, and $\mathscr{X}_1$ is the set of 1-labeled samples before modification. We denote $\mathcal{V}(\mathscr{X})$ the sum of all distinct one-to-one square distances in $\mathscr{X}$.

$$\mathcal{V}(\mathscr{X}) = \sum_{\mathbf{x}\in\mathscr{X}} \sum_{\mathbf{x}'\in\mathscr{X}} \|\mathbf{x}-\mathbf{x}'\|^2$$

Thanks to the adding-one-sample identity, we have

$$\mathcal{V}(\mathscr{X}_0' \bigcup \mathbf{x}) = \frac{N_0}{N_0-1}\mathcal{V}(\mathscr{X}_0') + 2(N_0-1)\|\mathbf{x}-\boldsymbol{\mu}_0'\|^2$$

and

$$\mathcal{V}(\mathscr{X}_1 \bigcup \mathbf{x}) = \frac{N_1+1}{N_1}\mathcal{V}(\mathscr{X}_1) + 2N_1\|\mathbf{x}-\boldsymbol{\mu}_1\|^2$$

The last question makes it possible to rewrite the first identity.

$$\mathcal{V}(\mathscr{X}_0' \bigcup \mathbf{x}) = \frac{N_0}{N_0-1}\mathcal{V}(\mathscr{X}_0') + 2N_0\|\mathbf{x}-\boldsymbol{\mu}_0\|^2$$

Answer to exercise 11 III

The definition of J and J' tells us

$$J = \frac{1}{2N_0}\mathcal{V}(\mathscr{X}_0' \bigcup \mathbf{x}) + \frac{1}{2N_1}\mathcal{V}(\mathscr{X}_1)$$

$$J' = \frac{1}{2(N_0-1)}\mathcal{V}(\mathscr{X}_0') + \frac{1}{2(N_1+1)}\mathcal{V}(\mathscr{X}_1 \bigcup \mathbf{x})$$

We finally get

$$J' - J = \frac{N_1}{N_1+1}\|\mathbf{x}-\boldsymbol{\mu}_1\|^2 - \frac{N_0}{N_0-1}\|\mathbf{x}-\boldsymbol{\mu}_0\|^2$$

5. The label of $\mathbf{x}$ is changed from 0 to 1 in the kmeans-algorithm because $\|\mathbf{x}-\boldsymbol{\mu}_0\|$ is greater than $\|\mathbf{x}-\boldsymbol{\mu}_1\|$. So here we have

$$\|\mathbf{x}-\boldsymbol{\mu}_1\|^2 \leq \|\mathbf{x}-\boldsymbol{\mu}_0\|^2$$

We can then prove that $J' \leq J$ if we show that $\frac{N_1}{N_1+1} \leq \frac{N_0}{N_0-1}$ And the latter is true as

$$\frac{N_1}{N_1+1} - \frac{N_0}{N_0-1} = \frac{N_1(N_0-1)-(N_1+1)N_0}{(N_1+1)(N_0-1)} < 0$$

This proves also that if a 1-sampled label was replaced with a 0-sampled label for a similar reason, we would also have $J' \leq J$. To really complete the proof we would need to consider the case where

Answer to exercise 11 IV

multiple samples are relabeled and this is out of the focus of this lecture. In simulations it appears that J is also non-increasing.

Conclusion of subsection 6, Unsupervised classification regarded as an optimization problem

- Description of a very popular algorithm: kmeans
- It is an unsupervised algorithm
- There exists a loss function for which this algorithm is non-increasing
- In terms of algorithm efficiency, this property is an appealing characteristic, but it is far from explaining the generally good performance and its popularity.
- Knowing the equation of this loss function can be used to adapt this algorithm to other contexts.

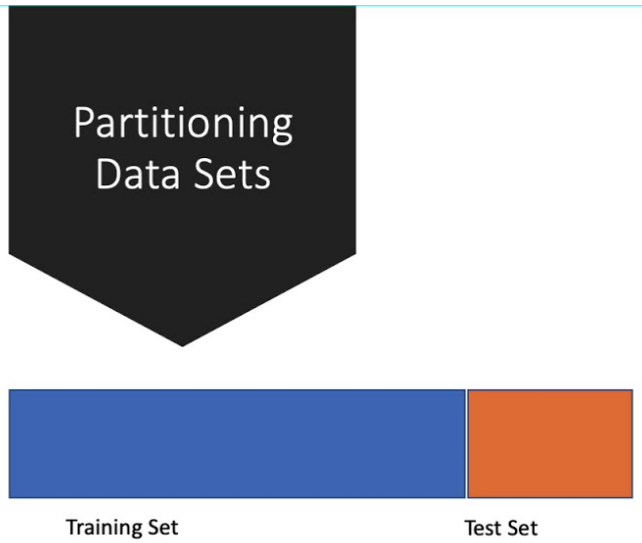
We have seen algorithms that seem to have good performance in terms of accuracy or at least with a loss function, can we say something about the reliability of a prediction regarding a new sample.

In the next section, we are measuring the reliability of such predictions?

Table of Contents

Content of section 3, Predicting the learning performances and probabilistic framework

Training and testing set



X(Input)	Y(Target)	
5	10	
10	18	Training Set
15	22	
18	27	
22	31	
25	35	
28	37	
32	41	Test Set
35	44	
39	48	

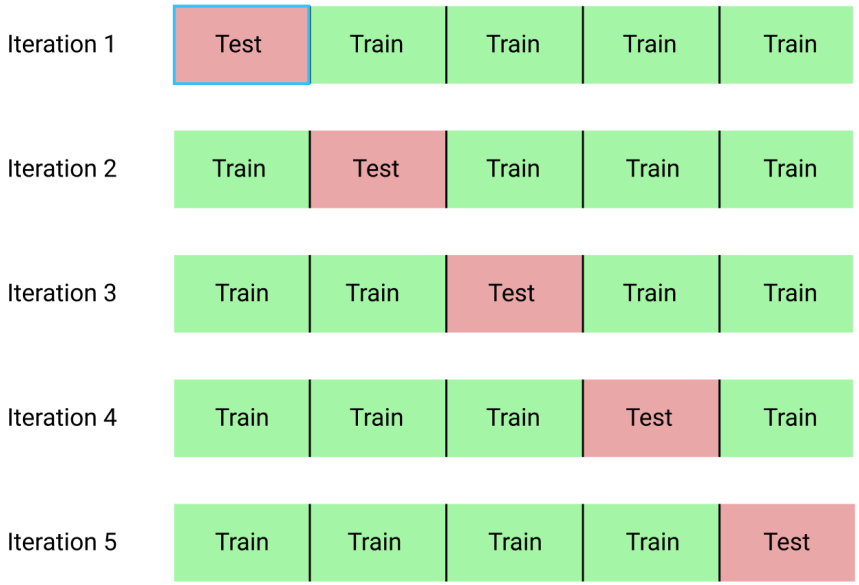
- Training set
- Test set
- Supervised classification problem

Validation set

- Different from Training set and Test set.
- Differences caused by Randomization and/or Overfitting
- Size could be of $\frac{1}{3}$ of the labeled samples available.
- Trade-off between reliability and scarcity of labeled samples.
- Ground truth is costly and could be erroneous.
- Numerical complexity could be an issue.

Cross-validation set

Use of validation sets to select among parameter values $\{\theta_1 \dots \theta_P\}$.
Example with $K = 5$.



$$\mathcal{A}_{k,p} = \text{TEST}(\text{LEARN}(\mathcal{S}_{k' \neq k}, \theta_p), \mathcal{S}_k)$$

$$\theta_{\text{opt}} = \theta_{p_{\text{opt}}} = \underset{p \leq P}{\text{argmin}} \sum_k \mathcal{A}_{k,p}$$

Exercise 12

Given a certain data set $\mathscr{S}_3 \bigcup \mathscr{S}_4$ with $\mathscr{S}_3$ as labeled and $\mathscr{S}_4$ not labeled.

1. *Improve the following algorithm using validation sets.*

Require: $\mathscr{S}_3, \mathscr{S}_4$: *data sets*
Ensure: $\mathbf{a}, b$: *linear classifier*

Compute $\mathscr{S}_{opt} = \mathscr{S}_3$.
$(\mathbf{a}_{opt}, b_{opt}) = LEARN(\mathscr{S}_{opt})$
Compute $\mathcal{A}_{opt}$ *with* $(\mathbf{a}_{opt}, b_{opt})$ *and* $\mathscr{S}_{opt}$.
repeat
    $(\mathbf{x}, (\mathbf{x}', y')) = \text{argmin}_{\mathbf{x} \in \mathscr{S}_4, (\mathbf{x}', y') \in \mathscr{S}_3} d(\mathbf{x}', \mathbf{x})$
    Set $\mathscr{S} = \mathscr{S}_{opt} \bigcup (\mathbf{x}, y')$
    $(\mathbf{a}, b) = LEARN(\mathscr{S})$
    Compute $\mathcal{A}$ *with* $(\mathbf{a}, b)$ *and* $\mathscr{S}$
    if $\mathcal{A} > \mathcal{A}_{opt}$ **then**
        $(\mathbf{a}_{opt}, b_{opt}) = (\mathbf{a}, b)$, $\mathscr{S}_{opt} = \mathscr{S}$, $\mathcal{A}_{opt} = \mathcal{A}$.
until $\mathcal{A} <= \mathcal{A}_{opt}$

Answer to exercise 12

Require: $\mathscr{S}_3, \mathscr{S}_4, I$

Ensure: $[(\mathbf{a}, b), \mathcal{A}] = \text{LEARN}(\mathscr{S}_3, \mathscr{S}_4, I)$

1: Set $\mathscr{S}_{\text{opt}}$, $(\mathbf{a}_{\text{opt}}, b_{\text{opt}})$, $\mathcal{A}_{\text{opt}}$.
2: **for** $i = 1 : I$ **do**
3: $(\mathbf{x}, (\mathbf{x}', y')) = \text{argmin}_{\mathbf{x} \in \mathscr{S}_4, (\mathbf{x}', y') \in \mathscr{S}_3} d(\mathbf{x}', \mathbf{x})$
4: Set $\mathscr{S} = \mathscr{S}_{\text{opt}} \bigcup (\mathbf{x}, y')$
5: $(\mathbf{a}, b) = \text{LEARN}(\mathscr{S})$
6: Compute $\mathcal{A}$ with $(\mathbf{a}, b)$ and $\mathscr{S}$
7: **if** $\mathcal{A} > \mathcal{A}_{\text{opt}}$ **then**
8: $(\mathbf{a}_{\text{opt}}, b_{\text{opt}}) = (\mathbf{a}, b)$, $\mathscr{S}_{\text{opt}} = \mathscr{S}$, $\mathcal{A}_{\text{opt}} = \mathcal{A}$.

Continuation of answer to exercise 12

Require: $\mathscr{S}_3, \mathscr{S}_4$
Ensure: $(\mathbf{a}, b)$

1: $\mathscr{S}_{3k} = \text{SPLIT}(\mathscr{S}_3, K)$
2: **for** $i = 1 : I$ **do**
3: $\quad \mathcal{A}_i = 0$
4: $\quad$ **for** $k = 1 : K$ **do**
5: $\quad\quad \mathcal{A}_i = \mathcal{A}_i + \text{LEARN}(\mathscr{S}_3, \mathscr{S}_4, i)/K$
6: $i_{\text{opt}} = \underset{i}{\text{argmax}} \mathcal{A}_i$
7: $[(\mathbf{a}, b), \mathcal{A}] = \text{LEARN}(\mathscr{S}_3, \mathscr{S}_4, i_{\text{opt}})$

Conclusion of subsection 1, Training, testing and validation sets I

- The question of the training set, validation set and testing set, is generally studied in the context of supervised learning (labeled samples).
- We have seen the definitions of training, validation and test set and the cross validation technique.
- When we study a technique and want to assess its performance we need to now the true labels of the test samples.
- In a given application, we would be using the technique on samples for which we don't know the true label and we would give some confidence in the prediction yielded by the technique.
- The use of a validation set and of the cross validation technique are precisely tools that can tell us more specifically what confidence we may have.

Conclusion of subsection 1, Training, testing and validation sets II

- Regarding the unsupervised learning, we could build similarly the same sets. We can also consider that samples from the test set can be used to increase or update the knowledge we have.

Confusion matrix?

In the next section in order to study the reliability of a given technique based on its performance on a training set, we need a more precise indicator to describe the obtained performances, better than accuracy.

Content of section 3, Predicting the learning performances and probabilistic framework

Confusion matrix

Predicted Labels $\longrightarrow$

True Labels $\downarrow$

$$\mathbf{C} = \begin{bmatrix} \sum_{n=1}^{N} \delta(y_n = \widehat{y}_n = 0) & \sum_{n=1}^{N} \delta(y_n = 0 \text{ and } \widehat{y}_n = 0) \\ \sum_{n=1}^{N} \delta(y_n = 1 \text{ and } \widehat{y}_n = 0) & \sum_{n=1}^{N} \delta(y_n = \widehat{y}_n = 1) \end{bmatrix}$$

Exercise 13

We consider the following confusion matrix.

$$C = \begin{bmatrix} 5,1 \\ 1,5 \end{bmatrix}$$

1. *Give an example of* Y *and* $\widehat{Y}$ *consistent with* $\mathbf{C}$.
2. *Given* $Y^T = [0,0,0,0,0,0,1,1,1,1,1,1]$, *how many different* $\widehat{Y}$ *are consistent with* $\mathbf{C}$*?*

Answer to exercise 13

1. $$\widehat{Y}^{\mathsf{T}} = [1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0]$$
2. 6×6.

$$[1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1]$$
$$\vdots$$
$$[1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0]$$
$$[0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0]$$
$$\vdots$$
$$\vdots$$
$$[0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0]$$

Conclusion of subsection 2, Confusion matrix

- We have seen the definition of the confusion matrix
- It should not be confused the transpose of this confusion matrix. When go down, scrolling down the different rows, we get information on samples having actually different labels. When going to the right, we get information on samples having different predicted labels.
- In non-binary classification problems, confusion matrix are not of size 2×2.

How are the confusion matrix going to be used in the next section?

We are considering different experiments for which techniques have parameters yielding a performance measured by a unique confusion matrix. So we are studying what we can see differences that are not measured by confusion matrices.

Content of section 3, Predicting the learning performances and probabilistic framework

A linear classifier separating gaussians

Exercise 14

Let Y be a uniform binary random variable and X when conditioned to Y be a 2D-gaussian variable with mean $\boldsymbol{\mu}_0 \in \mathbb{R}^2$ or $\boldsymbol{\mu}_1 \in \mathbb{R}^2$ and standard deviation $\sigma_0 > 0$ or $\sigma_1 > 0$.

1. *What is the probability that $Y = 0$ on a given experiment?*
2. *What is the probability density function that $X = [x_1, x_2]$ given $Y = 0$ and then given $Y = 1$?*
3. *We now assume that $\sigma_0 = \sigma_1 = \sigma$, show that a straight line separates points that are more likely when $Y = 1$ from the more likely points when $Y = 0$.*

$$f_{X|Y=1}(\mathbf{x}) \geq f_{X|Y=0}(\mathbf{x}) \Leftrightarrow (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\mathbf{x}^T \geq (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)(\frac{1}{2}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0)^T$$

The last question refers to an example of linear discriminant analysis that we will discuss at the end of this section.

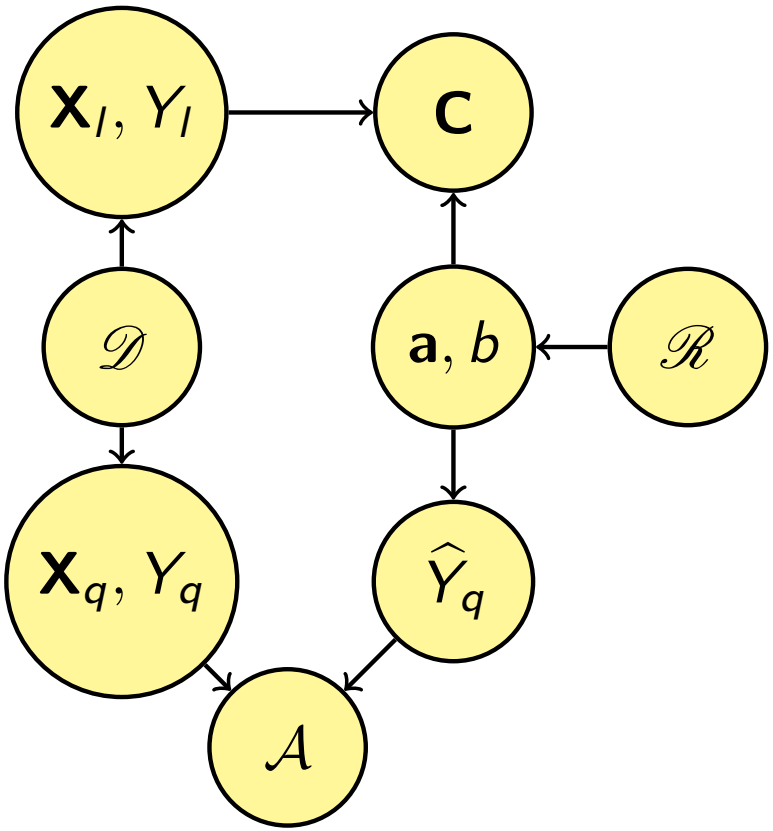
Answer to exercise 14

1. $$\begin{cases} P(Y=0) = P(Y=1) \\ P(Y=0) + P(Y=1) = 1 \end{cases} \Rightarrow P(Y=0) = 0.5$$

2. $$f_{X|Y=0}(\mathbf{x}) = \frac{1}{2\pi\sigma_0^2} e^{-\frac{1}{2\sigma_0^2}(\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^\mathsf{T}}$$
$$f_{X|Y=1}(\mathbf{x}) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{1}{2\sigma_1^2}(\mathbf{x}-\boldsymbol{\mu}_1)(\mathbf{x}-\boldsymbol{\mu}_1)^\mathsf{T}}$$

3. $$\begin{aligned} \frac{f_{X|Y=1}(\mathbf{x})}{f_{X|Y=0}(\mathbf{x})} &= e^{\frac{1}{2\pi\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^\mathsf{T} - \frac{1}{2\pi\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_1)(\mathbf{x}-\boldsymbol{\mu}_1)^\mathsf{T}} \geq 1 \\ &\Leftrightarrow (\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^\mathsf{T} \geq (\mathbf{x}-\boldsymbol{\mu}_1)(\mathbf{x}-\boldsymbol{\mu}_1)^\mathsf{T} \\ &\Leftrightarrow -2\boldsymbol{\mu}_0\mathbf{x}^\mathsf{T} + \boldsymbol{\mu}_0\boldsymbol{\mu}_0^\mathsf{T} \geq -2\boldsymbol{\mu}_1\mathbf{x}^T + \boldsymbol{\mu}_1\boldsymbol{\mu}_1^\mathsf{T} \\ &\Leftrightarrow (\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)\mathbf{x}^\mathsf{T} \geq (\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)(\tfrac{1}{2}\boldsymbol{\mu}_0 + \tfrac{1}{2}\boldsymbol{\mu}_1)^\mathsf{T} \end{aligned}$$

An experiment



$\mathbf{a}, b$ are randomly chosen according to $\mathscr{R}$.

$\mathbf{x}$ are drawn according to a distribution $\mathscr{D}$.

Training set:12 samples

$$Y_l^T = [0,0,0,0,0,0,1,1,1,1,1,1]$$

$$\widehat{Y}_l^T = [1,0,0,0,0,0,1,1,1,1,1,0]$$

Confusion matrix

$$C = \begin{bmatrix} 5,1 \\ 1,5 \end{bmatrix}$$

Testing set: 2 samples

$$Y_q^T = [0,1]$$

Accuracy: 3 possible values

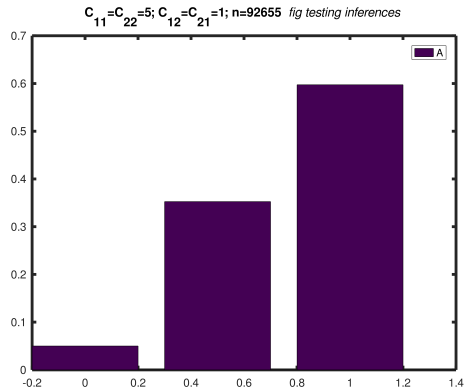
$$A = \frac{1}{2}\delta(y_{q0} = \widehat{y}_{q0}) + \frac{1}{2}\delta(y_{q1} = \widehat{y}_{q1})$$

Algorithm of a random classifier

Require: $\mathbf{C}$

Ensure: $P(\mathcal{A})$

Set $P(\mathcal{A}) = [0, 0, 0]$.
for $i = 1 : I$ **do**
  repeat
    Draw $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \sigma_0, \sigma_1$, $\mathbf{a}$ and b.
    Set $Y_I^T = [0 \ldots 0, 1 \ldots 1]$.
    Draw $\mathbf{X}_I$.
    Compute $\widehat{Y}_I$ with $\mathbf{X}_I$ and $\widehat{C}$ with $Y_I, \widehat{Y}_I$.
  until $\widehat{\mathbf{C}} = \mathbf{C}$
  Set $Y_q^T = [0, 1]$.
  Draw $\mathbf{X}_q$.
  Compute $\widehat{Y}$
  Compute $\mathcal{A} = \frac{1}{2}\delta(\widehat{y}_{q0} = 0) + \frac{1}{2}\delta(\widehat{y}_{q1} = 1)$
  Adapt $P(\mathcal{A})$ with $\mathcal{A}$
Normalize $P(\mathcal{A})$



Conditional probabilities

$P(\mathcal{A} = 0 | \widehat{C} = C)$,
$P(\mathcal{A} = 0.5 | \widehat{C} = C)$,
$P(\mathcal{A} = 1 | \widehat{C} = C)$

We assume here it is very unlikely that $\widehat{C} = C$

- $P(\mathcal{A} = 1 \text{ and } \widehat{C} = C)$ means the probability of having $\mathcal{A} = 1$ **and** that $\widehat{C} = C$
- The assumption implies $P(\mathcal{A} = 1 \text{ and } \widehat{C} = C)$ is small.
- If each time $\widehat{C} = C$, we also have $\mathcal{A} = 1$ then the assumption makes it invisible in $P(\mathcal{A} = 1 \text{ and } \widehat{C} = C)$

- $P(\mathcal{A} = 1 | \widehat{C} = C)$ means the probability of having $\mathcal{A} = 1$ **given** that $\widehat{C} = C$
- The assumption does not imply anything on $P(\mathcal{A} = 1 | \widehat{C} = C)$
- If each time $\widehat{C} = C$, we also have $\mathcal{A} = 1$ then $P(\mathcal{A} = 1 | \widehat{C} = C) = 1$ is high.

Example on the computation of conditional probabilities

Concerning a dice, we consider an event *E dice equal 1* and a side information *S dice is odd*.

Two theoretical formulas

First definition

Second definition

$$P(E|S) = \frac{P(E\&S)}{P(S)}$$

dice	E	S
1	1	1
2	0	0
3	0	1
4	0	0
5	0	1
6	0	0

```
dice=ceil(rand(1,1000)*6);

odd=@(n)mod(n,2)==1;
dice2=dice(odd(dice));

proba_EGS_1=sum(dice2==1)/length(dice2),

proba_E=sum(mod(dice,2)==1)/length(dice),
proba_S=sum(dice2==1)/length(dice),
proba_EGS_2=proba_E/proba_S,
```

Conclusion of subsection 3, Inference on an example

- By repeating a random experiment, we can measure inference.
- Probability distributions is a interesting framework to describe experiments.

As a side effect

From this probabilistic framework we get a new classifier.

Content of section 3, Predicting the learning performances and probabilistic framework

LDA on a simplified example

Exercise 15

We consider here a data set defined by a probability distribution.

$P(y=0) = P(y=1) = 0.5$ *and*

$$\begin{cases} f_{\mathbf{x}|y=0}(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^T} \\ f_{\mathbf{x}|y=1}(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_1)(\mathbf{x}-\boldsymbol{\mu}_1)^T} \end{cases}$$

with $\boldsymbol{\mu}_0 = [1,0], \boldsymbol{\mu}_1 = [0,1]$ *and* $\sigma = 2$.

1. *Write an algorithm to check that these expressions are probability distributions. Use the independence between the two components to reduce the numerical complexity.*
   $\int_{x_1}\int_{x_2} f(x_1)f(x_2)dx_1dx_2 = \int_{x_1} f(x_1)dx_1 \int_{x_2} f(x_2)dx_2$
2. *Show that with this model,* $y=1$ *is more likely than* $y=0$ *iff*
   $$\boldsymbol{\mu}_0\boldsymbol{\mu}_0^T - \boldsymbol{\mu}_1\boldsymbol{\mu}_1^T - (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)\mathbf{x}^T \geq 0$$
3. *Draw in the feature space the domains for which* $y=1$ *or* $y=0$ *is more likely.*

Answer to exercise 15 I

1. We need to check

$$\int_{x_1=-\infty}^{+\infty}\int_{x_2=-\infty}^{+\infty} f_{\mathbf{x}|y=0}(\mathbf{x})\,dx_1\,dx_2 = \int_{x_1=-\infty}^{+\infty}\int_{x_2=-\infty}^{+\infty} f_{\mathbf{x}|y=1}(\mathbf{x})\,dx_1\,dx_2 = 1$$

Require: σ, y

Ensure: s value of the integral

Set $s = 0$, $Q = 1e-2$
for $q_1 = -\frac{1}{Q^2} : \frac{1}{Q^2}$ **do**
    Set $x_1 = q_1 Q$
    for $q_2 = -\frac{1}{Q^2} : \frac{1}{Q^2}$ **do**
        Set $x_2 = q_2 Q$
        Add to s, $f_{\mathbf{x}|y}(x_1, x_2)Q^2$
Display s that should be close to 1

Answer to exercise 15 II

However this is actually quite complex. So we separate what happens to each component.

$$f_{\mathbf{x}|y=0}(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_1-\mu_{01})^2} \times \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_2-\mu_{02})^2}$$

$$f_{\mathbf{x}|y=1}(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_1-\mu_{11})^2} \times \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_2-\mu_{12})^2}$$

$$\int_{x_1=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_1-\mu_{01})^2} \int_{x_2=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_2-\mu_{02})^2} =$$

$$\int_{x_1=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_1-\mu_{11})^2} \int_{x_2=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_2-\mu_{12})^2} = 1$$

Require: σ, y

Ensure: s value of the integral

1: Set $s_1 = s_2 = 0$, $Q = 1e-2$
2: **for** $q_1 = -\frac{1}{Q^2} : \frac{1}{Q^2}$ **do**
3: Set $x_1 = q_1 Q$
4: Add to s_1, $f_{x_1|y}(x_1)Q$

Answer to exercise 15 III

5: **for** $q_2 = -\frac{1}{Q^2} : \frac{1}{Q^2}$ **do**
6: Set $x_2 = q_2 Q$
7: Add to s_2, $f_{x_2|y}(x_2)Q$
8: Compute $s = s_1 s_2$.
9: Display s that should be close to 1

2. The goal is to find where in the feature space $f_{\mathbf{x}|y=1}(\mathbf{x}) > f_{\mathbf{x}|y=0}(\mathbf{x})$.

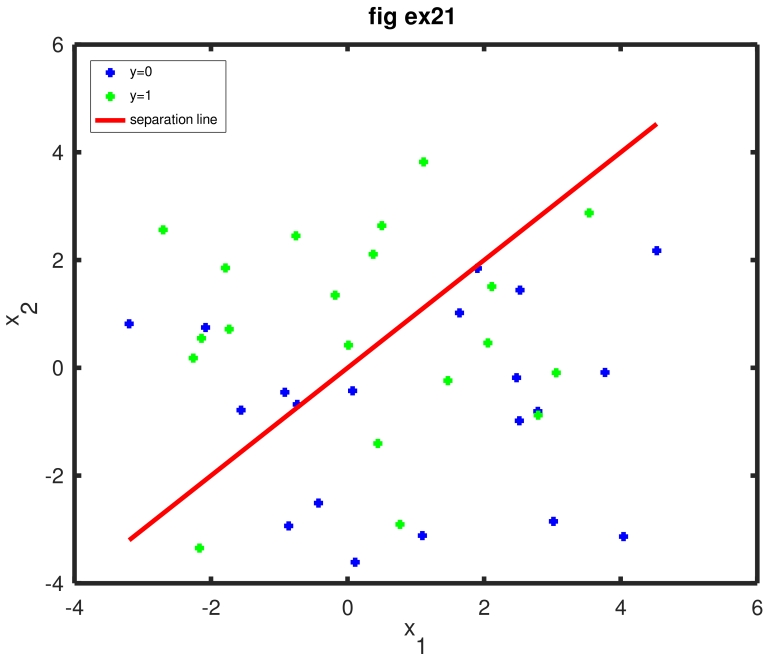
$$\sigma^2 \ln\left(\frac{f_{\mathbf{x}|y=1}(\mathbf{x})}{f_{\mathbf{x}|y=0}(\mathbf{x})}\right) = (\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^T - (\mathbf{x}-\boldsymbol{\mu}_1)(\mathbf{x}-\boldsymbol{\mu}_1)^T$$
$$= -2\boldsymbol{\mu}_0\mathbf{x}^T + \boldsymbol{\mu}_0\boldsymbol{\mu}_0^T + 2\boldsymbol{\mu}_1\mathbf{x}^T - \boldsymbol{\mu}_1\boldsymbol{\mu}_1^T$$

This proves $y = 1$ is more likely when

$$\boldsymbol{\mu}_0\boldsymbol{\mu}_0^T - \boldsymbol{\mu}_1\boldsymbol{\mu}_1^T - 2(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)\mathbf{x}^T \geq 0$$

Answer to exercise 15 IV

3. $y = 1$ is more likely when $x_2 \geq x_1$. Indeed
$$(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)\mathbf{x}^T = x_1 - x_2 \text{ and } \boldsymbol{\mu}_0\boldsymbol{\mu}_0^T - \boldsymbol{\mu}_1\boldsymbol{\mu}_1^T = 0$$



LDA in a more general context

Probabilistic assumption

$P(y = 1) = p = 1 - P(y = 0)$ and $f_{\mathbf{x}|y=1}(\mathbf{x})$ and $f_{\mathbf{x}|y=0}(\mathbf{x})$ are two independent multivariate normal distribution with an unknown **common covariance matrix** Σ.

$$f_{\mathbf{x}|y=1}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{F}{2}} |\det(\Sigma)|^{\frac{F}{2}}} e^{-(\mathbf{x}-\boldsymbol{\mu}_1)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)^T}$$
$$f_{\mathbf{x}|y=0}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{F}{2}} |\det(\Sigma)|^{\frac{F}{2}}} e^{-(\mathbf{x}-\boldsymbol{\mu}_0)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)^T}$$

Σ is defined as the covariance matrix

$$\Sigma = E\left[(\overset{r}{\mathbf{x}})^T \overset{r}{\mathbf{x}}\right]$$

where E is the expectation and here $\overset{r}{\mathbf{x}}$ is a random raw vector.

Covariance matrix

It is estimated with $\mathbf{X}$ from the training set.

$$\widehat{\Sigma} = \sum_{n=1}^{N} \mathbf{x}_n^T \mathbf{x}_n = \mathbf{X}^T \mathbf{X}$$

Note the striking similarity of this covariance matrix with $\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}$ used in the least square methodology.

Is it appropriate to assume a common covariance matrix?

This assumption yields a linear classifier. Besides it is generally difficult to estimate precisely Σ using all the samples in the training set, sometimes some regularization is needed to help the estimation. So it would be even more difficult to estimate two different covariance matrices.

Derived linear classifier

Similarly to exercise 15, we compute the logarithm of the ratio of

$$\begin{aligned}
&\ln f_{\mathbf{x}|y=1}(\mathbf{x}) - \ln f_{\mathbf{x}|y=0}(\mathbf{x}) \\
&= (\mathbf{x}-\boldsymbol{\mu}_0)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)^T - (\mathbf{x}-\boldsymbol{\mu}_1)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)^T \\
&= 2(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)\Sigma^{-1}\mathbf{x}^T \quad - \quad (\boldsymbol{\mu}_1\Sigma^{-1}\boldsymbol{\mu}_1^T - \boldsymbol{\mu}_0\Sigma^{-1}\boldsymbol{\mu}_0^T)
\end{aligned}$$

We get a linear classifier $f(\mathbf{x}) = \delta(b - \mathbf{a}.\mathbf{x} \geq 0)$ with

$$\begin{cases} \mathbf{a} = 2(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)\Sigma^{-1} \\ b = \boldsymbol{\mu}_0\Sigma^{-1}\boldsymbol{\mu}_0^T - \boldsymbol{\mu}_1\Sigma^{-1}\boldsymbol{\mu}_1^T \end{cases}$$

Supervised feature extraction

We could use $x' = b - \mathbf{a}.\mathbf{x}$ as an extracted feature. This is basically the idea behind some LDA-derived feature-extraction techniques. It is limited to the number of classes.

Conclusion of subsection 4, Linear discriminant analysis

We comparing with the L_2-linear classifier.

1. We also have to inverse the covariance matrix.
2. Instead of considering the cross-covariance matrix $\mathbf{X}^T Y$, we consider here distorted means, of 1-samples and 0-samples.
3. Just like L_2-linear classifier, it is prone to numerical instabilities when the covariance matrix is badly conditioned.

Question?

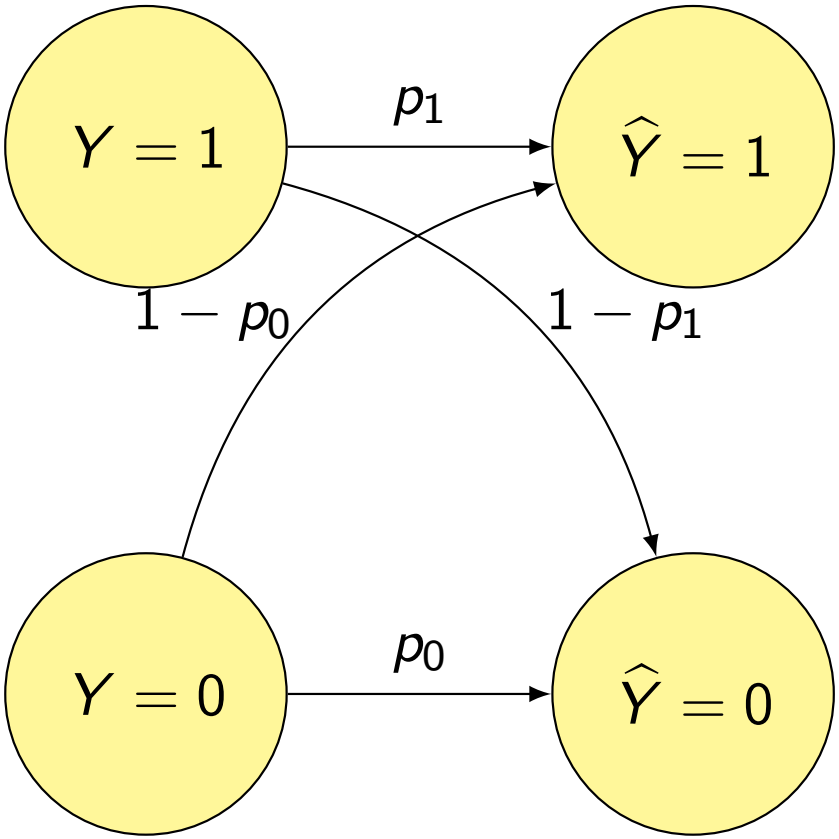
When applying this probabilistic framework to inference, can we make reliable predictions?

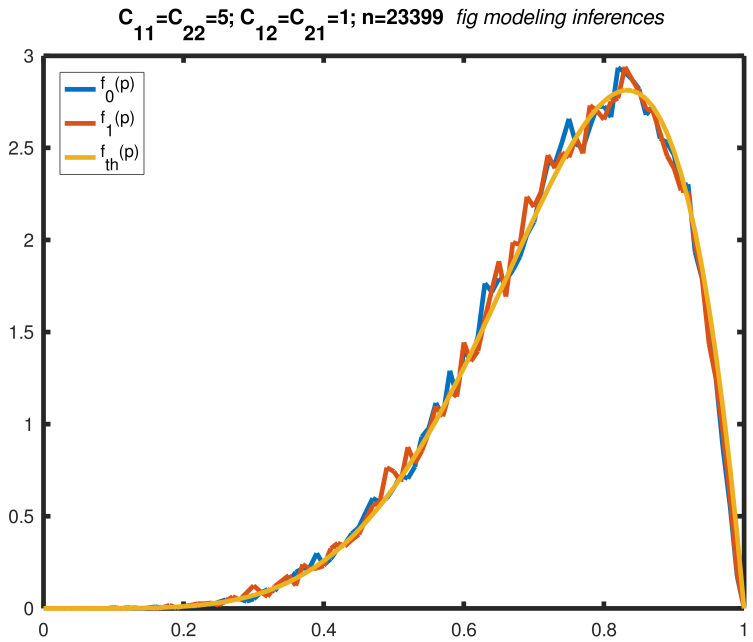
Content of section 3, Predicting the learning performances and probabilistic framework

Making inference on hidden parameters based on some evidence

It is common to compute the probability of having a given confusion matrix given a certain probabilistic model.
Here we do the opposite, get some probability on some parameters of a probabilistic model given that the observed confusion matrix meets some constraint.

Modeling the statistical inference





$$f_{th}(p) = \frac{p^5(1-p)}{\int_0^1 p^5(1-p)dp}$$

Exercise 16

We assume here an experiment of 12 samples, 6 labeled positively and 6 negatively. We observed for each label, that 5 of them are correctly predicted.

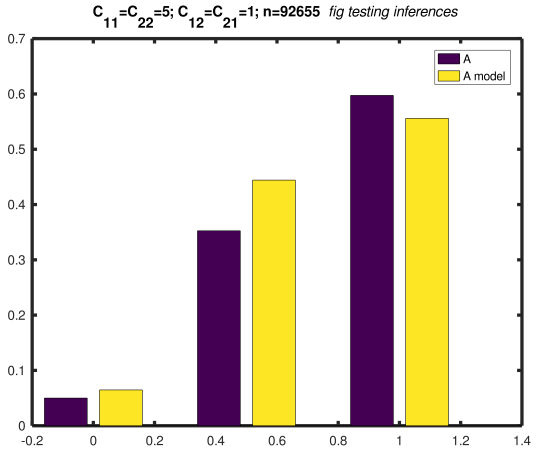
1. *Write an algorithm computing an approximation of the probability distributions that could best explain this experiment: the probability of a negative label to be correctly labeled $f_0(p)$ and that of a positive to be correctly labeled $f_1(p)$.*
2. *Given p_0 and p_1, and a column vector $Y^T = [0,0,0,0,0,0,1,1,1,1,1,1]$, show that the probability to have $\widehat{Y}$ consistent with the confusion matrix is*

$$\binom{6}{1} p_0^5 (1-p_0) \times \binom{6}{1} p_1^5 (1-p_1)$$

Answer to exercise 16

1. **Require:** $\mathbf{C}, Q, l$
   Ensure: $\mathbf{p}, \mathbf{f}_0, \mathbf{f}_1$
   1: Set $Y = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]$
   2: **for** $i = 1 : l$ **do**
   3: Draw p_0, p_1 as uniform variable on $[0, 1]$.
   4: Draw $\widehat{Y}$ along p_0 and p_1.
   5: Compute $\widehat{\mathbf{C}}$ according to $\widehat{Y}$ and Y.
   6: **if** $\widehat{\mathbf{C}} = \mathbf{C}$ **then**
   7: Adapt f_0 and f_1 with p_0 and p_1.
   8: Normalize f_0 and f_1.
2. What happens to the six first component is independent of the remaining. There are $\binom{6}{1} = 6$ ways of selecting a component in an array of 6 components. There is a probability of respectively p_0, p_1 to predict the correct value 0, 1, and $1 - p_0$, $1 - p_1$ to predict the incorrect values 1, 0.

$P(\mathcal{A}|\widehat{C} = C)$ are measured with two different techniques.



Comment on the figure

The second technique is a model of the first technique as any probabilistic model can be regarded as a random decision with some probability distribution for p_0 and p_1. Both distributions appear similar but they are not equal. Could we explain the difference?

- The technique shown in purple, draws randomly several multivariate normal distributions and measures $P(\mathcal{A}|\widehat{C} = C)$ by selecting only the instances where C is as expected.
- The technique shown in yellow, draws randomly some probabilities p_0 and p_1 of binary decisions and again only the accuracies corresponding to the expected C matrix are taken into account to compute $P(\mathcal{A}|\widehat{C} = C)$.

Conclusion of subsection 5, Predicting the true probabilities

1. We have modeled classifying samples as a binomial trial.
2. The confusion matrix measured during training yields the parameters of the binomial trial.
3. Our model yield a prediction accuracy.
4. Unfortunately it is not accurate.

How could we be more precise

We are going to consider the Bayesian framework with which the parameters of the binomial trial are regarded as random variables.

Content of section 3, Predicting the learning performances and probabilistic framework

Modeling a prior

- Prior is opposed to the posterior probability distribution.
- **Prior** refers to the assumed probability distribution before some evidence is given. Often the chosen probability distribution is the most general given some constraints.
- Here we know the experimental setup and we can test it without applying to data to read a probability distribution.

Require:
Ensure: Probability distribution of p_0 and p_1

1: **for** $i = 1 : l$ **do**
2: Draw $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \sigma_0, \sigma_1$, **a** and b.
3: Set $Y_l^T = [0 \dots 0, 1 \dots 1]$.
4: Draw $\mathbf{X}_l$.
5: Compute $\widehat{Y}_l$ with $\mathbf{X}_l$
6: Compute p_0 and p_1 by comparing $\widehat{Y}_l$ and Y_l.

Do we need a prior to compute a conditional probability?

No

Computing $P(C|p_0, p_1)$ does not require any prior. A specific value p_0, p_1 with the statistical model tells us the whole knowledge.

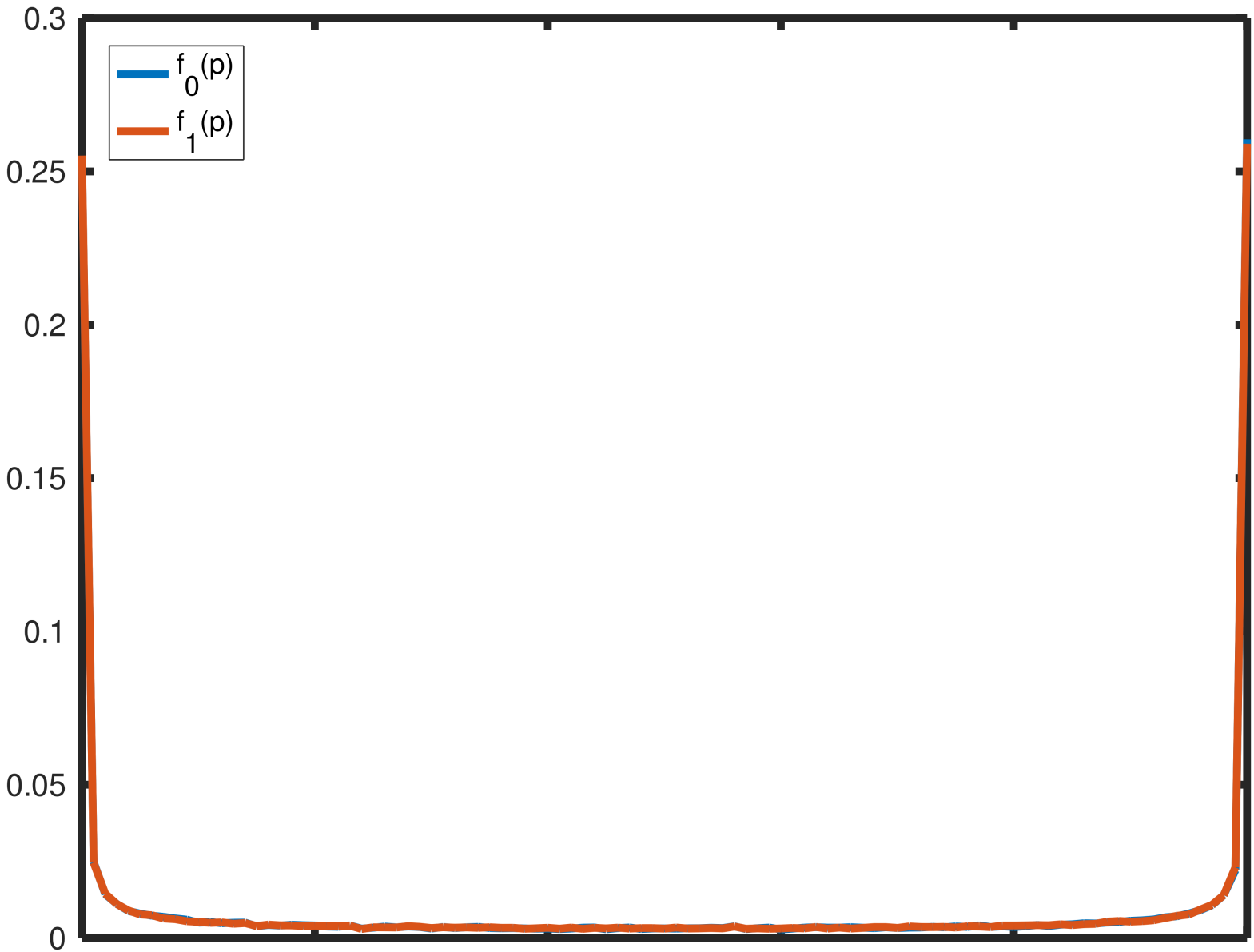
Yes

To compute $P(p_0, p_1|C)$ we consider all possible values of p_0 and p_1 and for each compute a probability of $P(C|p_0, p_1)$ and by counting the number of draws for which C has the appropriate value we get a probability of p_0, p_1. But the relative importance of p_0, p_1 is precisely a **prior**. In exercise 16, $p_0 np_1$ are drawn according to a **uniform distribution**.

We may not care

To what extent the choice of the prior is significant and appropriate are difficult questions. Not using it and considering that $P(p_0, p_1|C)$ and $P(C|p_0, p_1)$ are proportionate is actually a choice of prior that might be a not too bad choice.

Measured prior



Using Bayes formula to compute a posterior probability

Bayes formula

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\neg A)P(\neg A)}$$

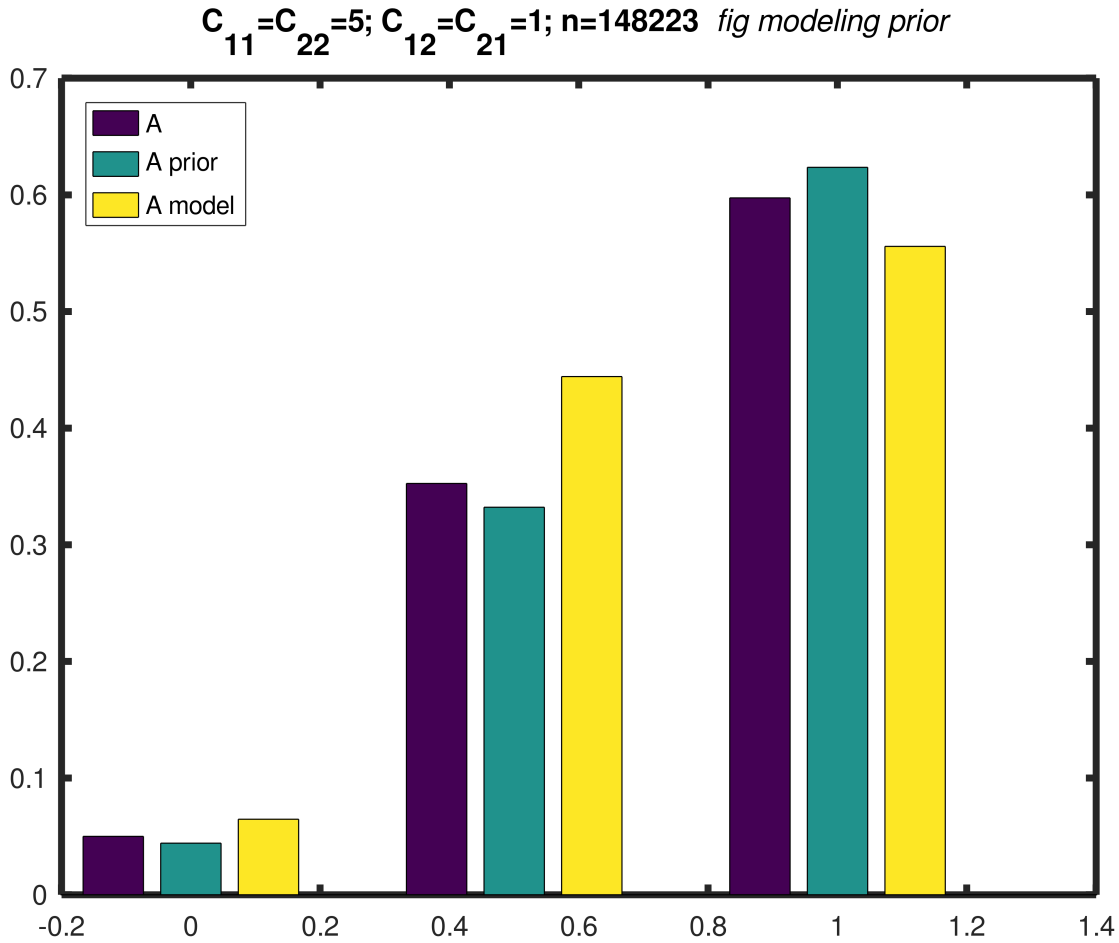
Applying this formula in our context

$$P(\mathcal{A} = a|\widehat{C} = C) = \int_{p_0,p_1} P(\mathcal{A} = a|\widehat{C} = C, p_0, p_1) f_0(p_0) f_1(p_1)\, dp_0\, dp_1$$

And we use for $f_0(p_0)$ and $f_1(p_1)$ the probability distribution measured without considering the C-constraints.

This posterior probability distribution of $\mathcal{A}$ is shown in green in the following figure.

Modeling with a prior



Because the green distribution is closer to the purple distribution, it seems that the prior is here useful.

Posterior probability vs maximizing the likelihood

The two viewpoints exist in the literature.

- Unknown parameters could have any value.
- It could be more precise.

- Unknown parameters are estimated taking into account the data.
- It makes computation easier and is often a good approximation.

Experiment using the maximum likelihood

Here we consider the most likely value of p_0 and p_1 that yield the expected C-matrix.

$$\underset{p}{\text{argmax}} f_{C|p}(p) = \underset{p}{\text{argmax}} p^5(1-p) =$$

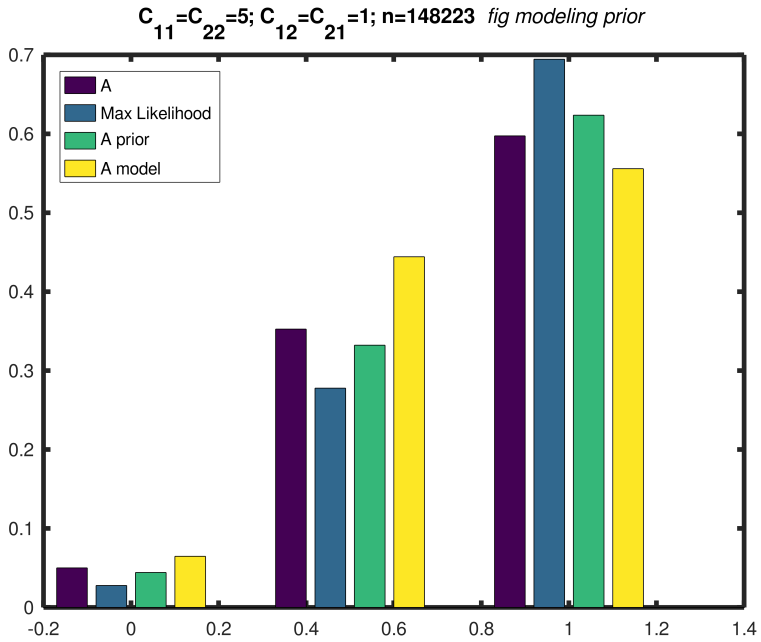
Since $\frac{d}{dp}p^5(1-p) = 0 \Rightarrow 5 - 6p = 0 \Rightarrow p = \frac{5}{6}$

We then get the distribution of $\mathcal{A}$

$$P(\mathcal{A}|\widehat{C} = C) =$$

$$P(\mathcal{A}|\widehat{C} = C, p_0 = p_1 = \tfrac{5}{6})$$

This new distribution of $\mathcal{A}$ is shown in blue.



Conclusion

Drawing adequate conclusions based on a certain success rate on the training set is definitely a hard issue.

Conclusion of section 3, Predicting the learning performances and probabilistic framework

1. In our attempt to have more precise predictions in terms of inference, we investigated the Bayesian framework.
2. Regarding an estimated parameter, rather than finding its best value, we assume it has an unknown value that follows a probability distribution.
3. This yields more precise predictions if the probability distribution is appropriate.

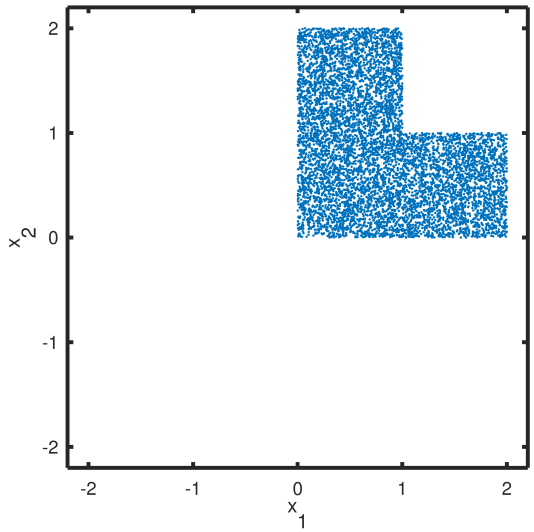
Conclusion

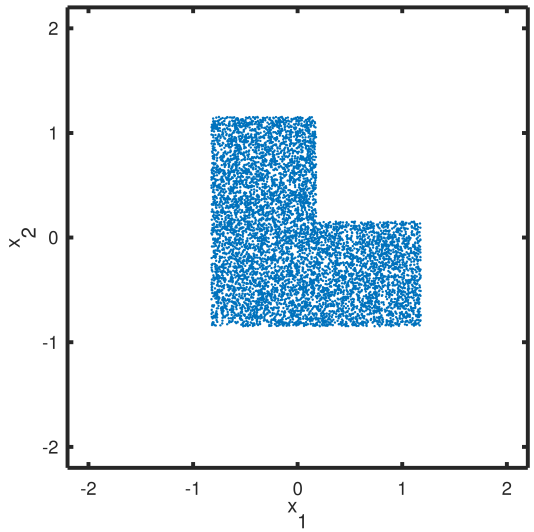
In my opinion, this framework is often relevant, it often increases accuracy sometimes by a very little amount, at the expense of an increased complexity.

Table of Contents

Content of section 4, Curse of dimensionality, regularization and sparsity

Centering the feature matrix





A centered feature matrix fulfills

$$\sum_{n=1}^{N} X_{n,f} = 0$$

Exercise 17

Let $\mathbf{X}$ *be a feature matrix. Show that there exists* β_f *such that* $\mathbf{X}' = \mathbf{X} - [\beta_1 \dots \beta_F]$ *is centered.*

Answer to exercise 17

Let β_f be defined as

$$\beta_f = \frac{1}{N}\sum_{n=1}^{N} X_{nf}$$

We then get for any $f \in \{1 \dots F\}$

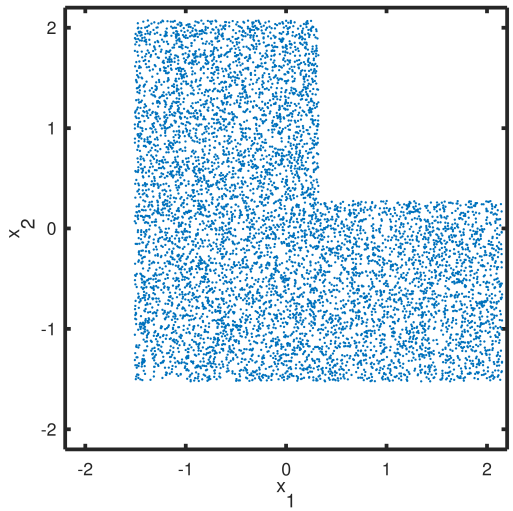
$$\sum_{n=1}^{N} X'_{nf} = \sum_{n=1}^{N}(X_{nf} - \beta_f) = \sum_{n=1}^{N} X_{nf} - \beta_f = 0$$

Normalizing features

Normalizing means

$$x_{nf} \mapsto x'_{nf} = \alpha_f x_{nf}$$

such that $\frac{1}{N}\sum_{n=1}^{N} x'^2_{nf} = 1$



Exercise 18

Given a data set $X = [x_{nf}]$, *compute a value* α_f *such that*

$$\frac{1}{N}\sum_{n=1}^{N} x'^2_{nf} = 1$$

where $x'_{nf} = \alpha_f x_{nf}$

Answer to exercise 18

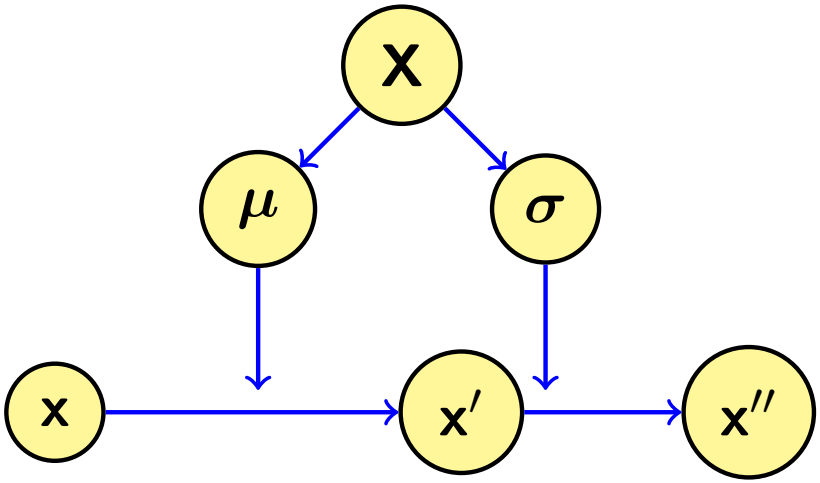
$$\alpha_f = \frac{1}{\sqrt{\frac{1}{N}\sum_{n=1}^{N} x_{nf}^2}}$$

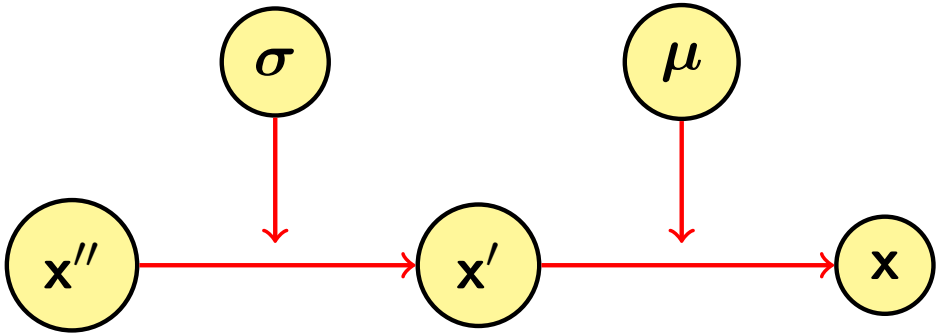
we get

$$\frac{1}{N}\sum_{n=1}^{N}(x'_{nf})^2 = \frac{1}{N}\sum_{n=1}^{N}\alpha_f^2 x_{nf}^2 = \alpha_f^2 \frac{1}{N}\sum_{n=1}^{N} x_{nf}^2 = \frac{\frac{1}{N}\sum_{n=1}^{N} x_{nf}^2}{\frac{1}{N}\sum_{n=1}^{N} x_{nf}^2} = 1$$

Analysis

Synthesis





Matrix notations

Exercise 19

The exercises 17 and 18 provided formulas to center and normalize the samples in the feature space. The goal here is to express these transformations with matrices. An interesting side-effect is the simplification of the implementation.
We consider here a dataset described with a matrix $\mathbf{X}$ *of size* $N \times F$ *and a column vector* Y *of size* $N \times 1$.

1. *Define a matrix* $\mathbf{H}$ *of size* $N \times N$ *such that* $\mathbf{HX}$ *is centered (i.e. the sums of each column of* $\mathbf{HX}$ *are null).*
2. *Show that* $\mathbf{HX}\left(diag(\mathbf{X}^T\mathbf{H}^2\mathbf{X})\right)^{-\frac{1}{2}}$ *is centered and normalized.*
3. *Write the Matlab/Octave implementation of* $\mathbf{HX}\left(diag(\mathbf{X}^T\mathbf{H}^2\mathbf{X})\right)^{-\frac{1}{2}}$

$$(diag(A))_{ij} = a_{ij}\delta(j=i) \text{ and } ((diag(A))_{ij})^{-\frac{1}{2}} = \frac{1}{\sqrt{a_{ii}}}\delta(j=i)$$

Matrix formulas

- Column number j of a matrix A:
$$\begin{bmatrix} a_{1j} \\ \vdots \\ a_{Ij} \end{bmatrix}$$
- Row number i of a matrix A:
$$[a_{i1}, \ldots, a_{iJ}]$$
- Left-multiplication of A by a diagonal matrix $D = [d_i \delta(j = i)]_{ij}$:
$$(DA)_{ij} = d_i a_{ij}$$
- Right-multiplication of A by a diagonal matrix $D = [d_i \delta(j = i)]_{ij}$:
$$(AD)_{ij} = a_{ij} d_j$$
- Multiplication of two matrices
$$(AB)_{ij} = \sum_k a_{ik} a_{kj}$$
- Left-multiplication of B by A^T
$$(A^T B)_{ij} = \sum_k a_{ki} a_{kj}$$

Answer to exercise 19 I

1. Let $\mathbf{H}$ be a $N \times N$ matrix defined as the identity matrix subtracted to a constant matrix equal to $\frac{1}{N}$

$$\mathbf{H} = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \dots & \vdots \\ 0 & \dots & 1 \end{bmatrix} - \frac{1}{N} \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix}$$

Components of $\mathbf{HX}$ are

$$(\mathbf{HX})_{ij} = x_{ij} - \frac{1}{N} \sum_{n=1}^{N} x_{nj}$$

The column number j is

$$\left(x_{1j} - \frac{1}{N} \sum_{n=1}^{N} x_{nj} \right), \dots, \left(x_{Fj} - \frac{1}{N} \sum_{n=1}^{N} x_{nj} \right)$$

Answer to exercise 19 II

2. Let $\mathbf{X}' = \mathbf{H}\mathbf{X}$. $\mathbf{X}'$ is centered.

$$\left(\mathbf{X}'^T\mathbf{X}'\right)_{ij} = \sum_{n=1}^{N} x'_{ni}x'_{nj}$$

$$\left(\text{diag}(\mathbf{X}'^T\mathbf{X}')\right)_{ij} = \sum_{n=1}^{N} (x'_{ni})^2\delta(j=i)$$

$$\left(\text{diag}(\mathbf{X}'^T\mathbf{X}')^{-\frac{1}{2}}\right)_{ij} = \frac{1}{\sqrt{\sum_{n=1}^{N} (x'_{ni})^2}}\delta(j=i)$$

$$\left(\mathbf{X}'\text{diag}(\mathbf{X}'^T\mathbf{X}')^{-\frac{1}{2}}\right)_{ij} = \frac{x'_{ij}}{\sqrt{\sum_{n=1}^{N} (x'_{nj})^2}}$$

Therefore $\mathbf{X}'\text{diag}(\mathbf{X}'^T\mathbf{X}')^{-\frac{1}{2}}$ is the centered and normalized matrix. And applying the transposing rules, we get

$$\mathbf{X}'\text{diag}(\mathbf{X}'^T\mathbf{X}')^{-\frac{1}{2}} = \mathbf{H}\mathbf{X}\text{diag}(\mathbf{X}^T\mathbf{H}^2\mathbf{X})^{-\frac{1}{2}}$$

3. `H=eye(N)-1/N*ones(N);`

```
Xp=H*X*diag(    diag(X'*H*H*X).^(-1/2)     );
```

Conclusion of subsection 1, Data preparation

Normalization gives equal importance to all features regardless of their variance.

Should we do centering and normalization?

Centering and normalization is generally considered a good practice. However, mean and standard deviation are not kept, it erases some information, this should be done considering the specific experiment.

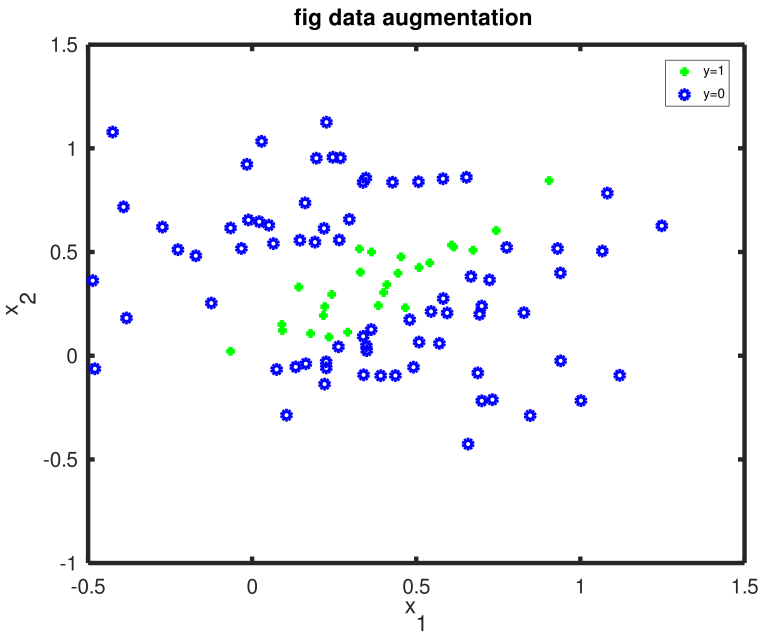
- If a feature variable has great variance (high value of $\frac{1}{N}\sum_{n=1}^{N} x_{nf}$), without normalization there is a high risk that only this variable is taken into account.
- If a feature variable contains only noise and has therefore little variance, normalization will give it more importance and data analysis could be compromised.

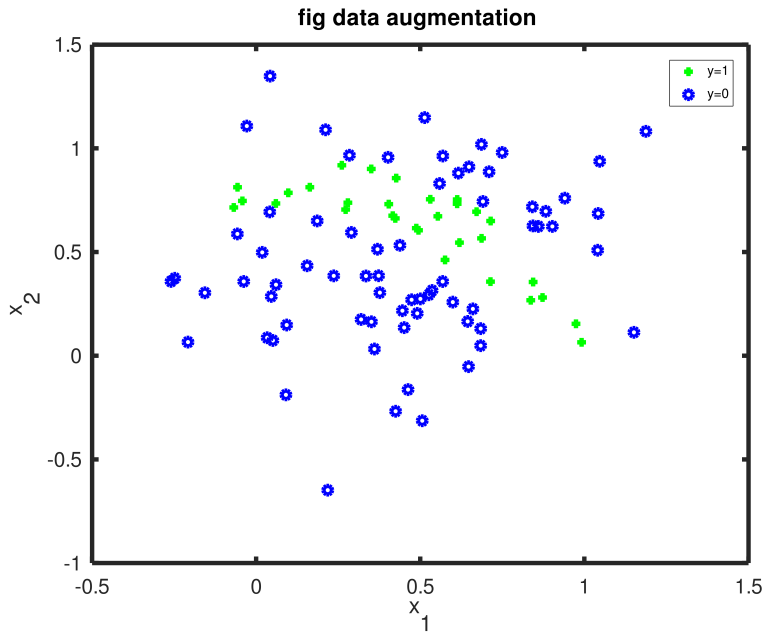
The given features can provide more information

Polynomial expansions

Content of section 4, Curse of dimensionality, regularization and sparsity

Can we classify the following datasets with a linear classifier?





Yes

With $\frac{F(F+1)}{2}$ new features:

$\{x_{f_1} x_{f_2} | f_1 \leq f_2\}$ here numbered with the lexicographic order

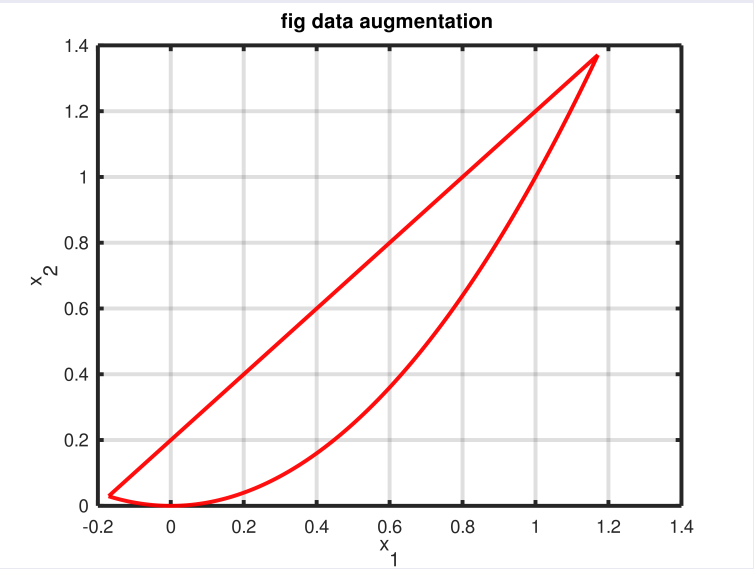
Notations

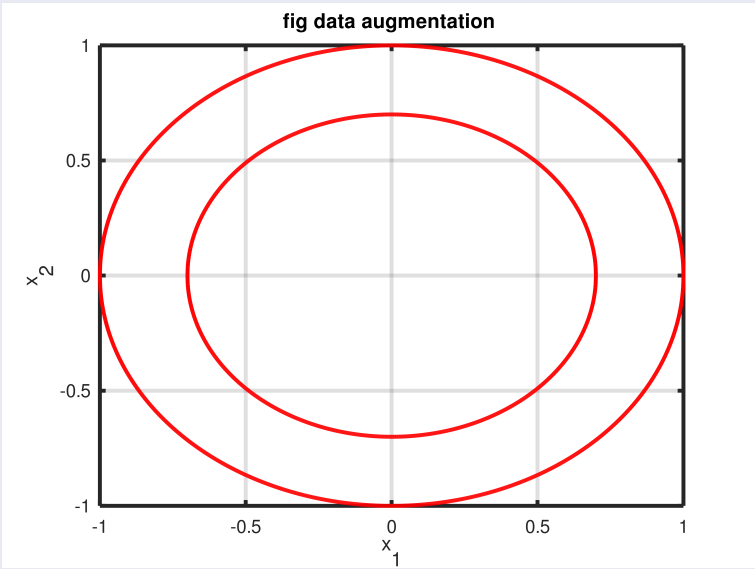
- $\mathbf{x}$ is a feature vector in the feature space $\mathscr{F}$.
- $\overset{\omega}{\mathbf{x}}$ is any feature vector in the augmented feature space denoted $\overset{\omega}{\mathscr{F}}$.

Examples of linear classifiers I

Exercise 20

The goal is to write linear classifiers corresponding to these domains in the feature space composed of two dimensions.





1. *Write equations delimiting the area of the left figure.*
2. *Write equations delimiting the area of the right figure.*
3. *Define the added features.*

Examples of linear classifiers II

Exercise

4. *Define two linear classifiers bounding the left area using also the added features.*

$$f(\overset{\omega}{\mathbf{x}}) = \delta(b_1 - \mathbf{a}_1 . \overset{\omega}{\mathbf{x}})\delta(b_2 - \mathbf{a}_2 . \overset{\omega}{\mathbf{x}})$$

with $f(\overset{\omega}{\mathbf{x}}) = 1$ *iff* $\mathbf{x}$ *is inside the domain.*

5. *Define two linear classifiers bounding the right area using also the added features.*

$$f(\overset{\omega}{\mathbf{x}}) = \delta(b_1 - \mathbf{a}_1 . \overset{\omega}{\mathbf{x}})\delta(b_2 - \mathbf{a}_2 . \overset{\omega}{\mathbf{x}})$$

with $f(\overset{\omega}{\mathbf{x}}) = 1$ *iff* $\mathbf{x}$ *is inside the domain.*

Answer to exercise 20 I

1. $$x_2 \leq x_1 + \frac{1}{5} \text{ and } x_2 \geq x_1^2$$
2. $$x_1^2 + x_2^2 \geq 0.7^2 \text{ and } x_1^2 + x_2^2 \leq 1$$
3. $F = 2$ and there are $\frac{F(F+1)}{2} = 3$ new features.
$$\overset{\omega}{x}_3 = x_1^2$$
$$\overset{\omega}{x}_4 = x_1 x_2$$
$$\overset{\omega}{x}_5 = x_2^2$$

Answer to exercise 20 II

4. The delimiting equations can be written as

$$\frac{1}{5} + \overset{\omega}{x}_1 - \overset{\omega}{x}_2 \geq 0$$

$$0 + \overset{\omega}{x}_2 - \overset{\omega}{x}_3 \geq 0$$

$$b_1 = \frac{1}{5} \quad \mathbf{a}_1 = [1, -1, 0, 0, 0]$$

$$b_2 = 0 \quad \mathbf{a}_2 = [0, 1, -1, 0, 0]$$

The delimiting equations can be written as

$$-0.7^2 + \overset{\omega}{x}_3 + \overset{\omega}{x}_5 \geq 0$$

$$1 - \overset{\omega}{x}_3 - \overset{\omega}{x}_5 \geq 0$$

$$b_1 = -0.7^2 \quad \mathbf{a}_1 = [0, 0, 1, 0, 1]$$

$$b_2 = 1 \quad \mathbf{a}_2 = [0, 0, -1, 0, -1]$$

We introduce some new notations

- $\omega(\mathbf{x})$ is the constructed feature vector.
- ω is a mapping of $\mathscr{F}$ into $\overset{\omega}{\mathscr{F}}$
  (i.e. injective or one-to-one but not surjective or onto and clearly not bijective or one-to-one correspondance).
- It is false to claim that $\forall \overset{\omega}{\mathbf{x}}, \exists \mathbf{x},\ \overset{\omega}{\mathbf{x}} = \omega(\mathbf{x})$.
- $\| \ \|$ is the Euclidean norm of $\mathscr{F}$ and $\| \ \|_\omega$ is the Euclidean norm of $\overset{\omega}{\mathscr{F}}$.

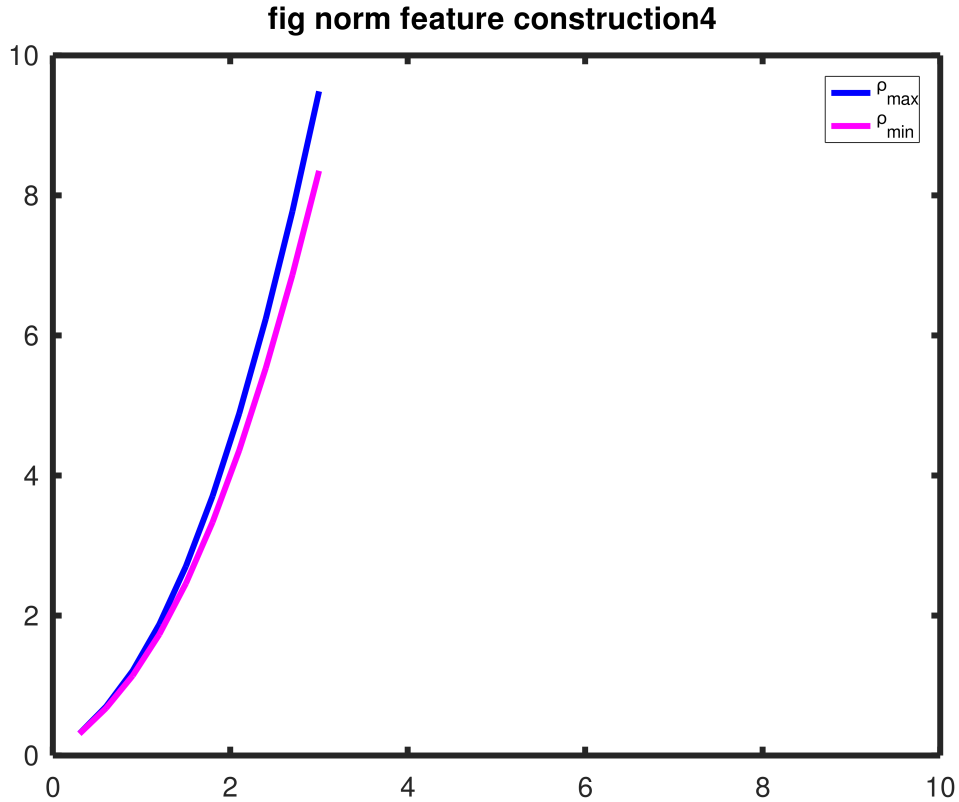
Contradiction between $\overset{\omega}{\mathscr{F}}$ and $\omega(\mathscr{F})$

The samples in the dataset is inside $\omega(\mathscr{F})$. However they are considered as members of the 5D-space denoted $\overset{\omega}{\mathscr{F}}$.

Growth of the distances

Generally when norms are compared we have some bounding properties: $\kappa_1 \leq \frac{\text{norm1}(x)}{\text{norm2}(x)} \leq \kappa_2$ Here we do not have this bounding property.

$$\|\mathbf{x}\|\sqrt{1+\frac{3}{4}\|\mathbf{x}\|^2} \leq \|\omega(\mathbf{x})\|_\omega \leq \|\mathbf{x}\|\sqrt{1+\|\mathbf{x}\|^2}$$

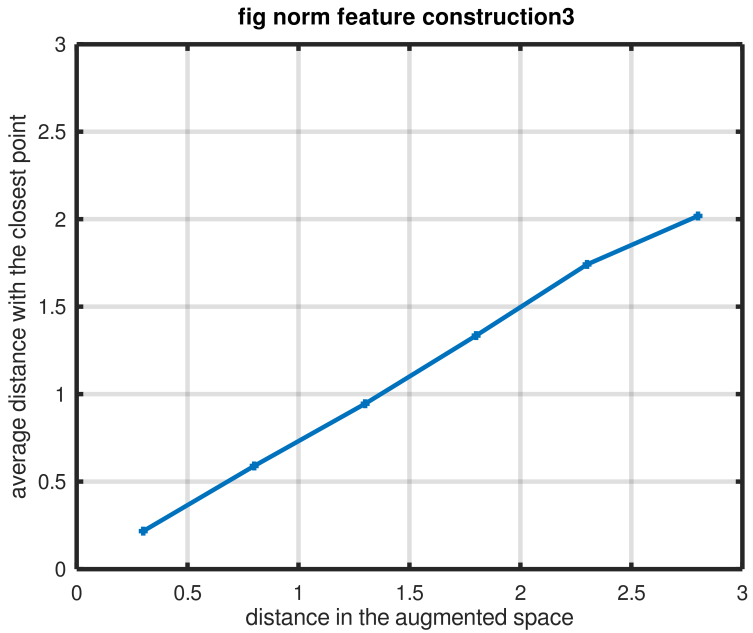


Most points in $\overset{\omega}{\mathscr{F}}$ are far from $\omega(\mathscr{F})$

Average distance between points in $\overset{\omega}{\mathscr{F}}$ and points that can be mapped from $\mathscr{F}$ with ω.

$$d(t) = E\left[\min_{\mathbf{x}' \in \mathscr{F}} \left\{ \|\omega(\mathbf{x}') - \overset{\omega}{\mathbf{x}}\|_\omega \,\middle|\, \|\overset{\omega}{\mathbf{x}}\|_\omega = t \right\}\right]$$

where E is expected value when following here the uniform law.

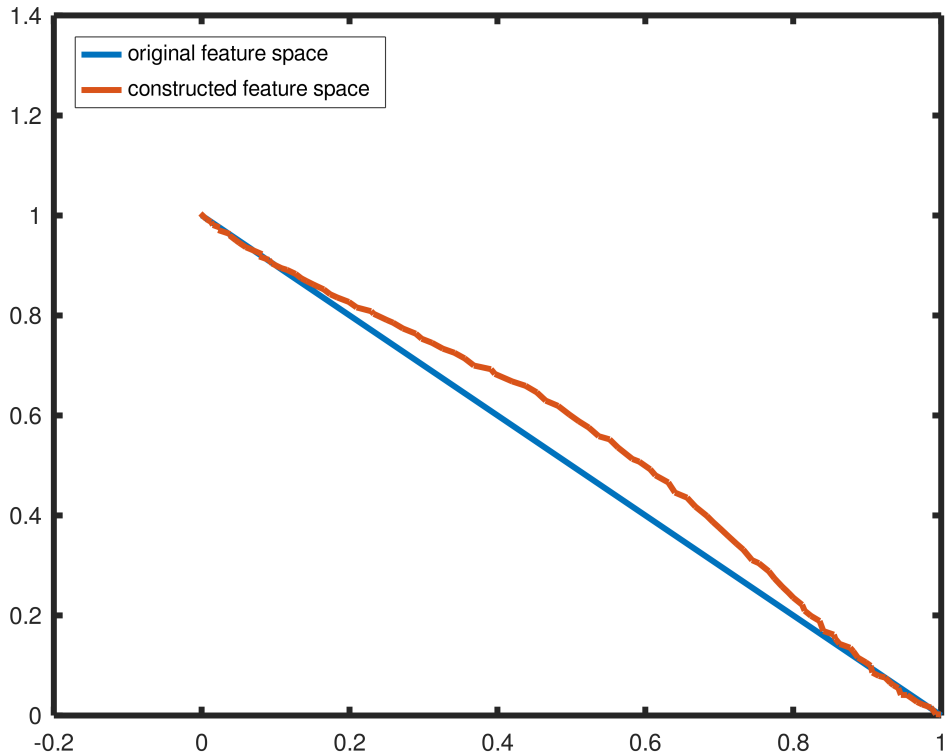


The closest point are close to where we expect them

We are considering a segment line in $\overset{\omega}{\mathscr{F}}$ joining two points in $\omega(\mathscr{F})$, $\omega(\mathbf{x}_1)$ and $\omega(\mathbf{x}_2)$. And we look for points $\mathbf{x}'$ in $\mathscr{F}$ which are mapped into the closest points of the segment line.

$$\mathbf{x}_\alpha = \underset{\mathbf{x}' \in \mathscr{F}}{\arg\min} \|\alpha\omega(\mathbf{x}_1) + (1-\alpha)\omega(\mathbf{x}_2) - \omega(\mathbf{x}')\|_\omega$$

fig geodesic



with $\alpha \in [0, 1]$

Conclusion of subsection 2, Feature construction

- Nonlinear transformations on features can transform a linear classifier into a more complex and possibly more appropriate classifier.
- We have studied the example of quadratic classifier.
- The extended feature space is embedded into a vector space but
  - $\| \ \|_\omega$ is different in nature from $\| \ \|$
  - $\| \ \|$ is different in value from $\|\omega(\ )\|_\omega$
  - Most points in the embedded feature space are far from the extended feature space
  - The projected points from the embedded space are not exactly where one might expect.

Reducing dimensions?

Content of section 4, Curse of dimensionality, regularization and sparsity

An example with $\omega(\mathscr{F})$ I

Exercise 21

We consider a small dataset

$$\mathbf{x}_1 = [1,\ 0]$$
$$\mathbf{x}_2 = [0,\ 1]$$
$$\mathbf{x}_3 = [1,\ 1]$$

We consider three new features X_1^2, x_1x_2 *and* x_2^2 *and its corresponding mapping* ω. *We consider a first kernel* $\mathcal{K}$

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \omega(\mathbf{x}).\omega(\mathbf{x}')$$

1. *Express* $\mathcal{K}$ *as function of* $[x_1, x_2]$ *and* $[x_1', x_2']$. *Is it left-linear, right-linear?*
2. *Compute* $\mathbf{K} = [\mathcal{K}(\mathbf{x}_m, \mathbf{x}_n)]_{m,n}$
3. *Show that the inverse of* $\mathbf{K}$ *is defined?*

An example with $\omega(\mathscr{F})$ II

Exercise

The inverse of $\mathbf{K}$ *is*

$$K^{-1} = \begin{bmatrix} 1.5 & 1 & -1 \\ 1 & 1.5 & -1 \\ -1 & -1 & 1 \end{bmatrix}$$

We define

$$\mathcal{K}(\mathbf{x}) = [\mathcal{K}(\mathbf{x}, \mathbf{x}_1), \mathcal{K}(\mathbf{x}, \mathbf{x}_2), \mathcal{K}(\mathbf{x}, \mathbf{x}_3)]\mathbf{K}^{-1} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}$$

4. *Compute* $\mathcal{K}(\mathbf{x}_1)$, $\mathcal{K}(\mathbf{x}_2)$ *and* $\mathcal{K}(\mathbf{x}_3)$.
5. *Show that there exists* $\mathbf{x}$ *such that* $\omega(\mathbf{x}) \notin \text{span}(\omega(\mathbf{x}_1), \omega(\mathbf{x}_2), \omega(\mathbf{x}_3))$. *Explain how we could manage to avoid this problem?*
6. *Compute* $\mathcal{K}(\mathbf{x}_1 - \mathbf{x}_2)$.

Answer to exercise 21 I

1. $$\mathcal{K}(\mathbf{x}, \mathbf{x}') = x_1 x_1' + x_2 x_2' + x_1^2 {x_1'}^2 + x_1 x_1' x_2 x_2' + x_2^2 {x_2'}^2$$
   It is not left-linear (nor right-linear for the same reasons). If it were then for $\mathbf{x}' = [1\ 0]$, the mapping $x_1 \mapsto x_1 + x_1^2$ would be linear.
2. $(\mathbf{K})_{11} = \mathcal{K}([1\,0], [1\,0]) = 1 \times 1 + 0 + 1^2 \times 1^2 + 0 + 0$
   $(\mathbf{K})_{12} = \mathcal{K}([1\,0], [0\,1]) = 1 \times 0 + 0 \times 1 + 1^2 \times 0^2 + 1 \times 0 \times 0 \times 1 + 0^2 \times 1^2$

$$\mathbf{K} = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 5 \end{bmatrix}$$

$\mathbf{K}$ is invertible because $\det(\mathbf{K}) \neq 0$.

$$\det(\mathbf{K}) = 2 \begin{vmatrix} 2 & 2 \\ 2 & 5 \end{vmatrix} + 2 \begin{vmatrix} 0 & 2 \\ 2 & 2 \end{vmatrix} = 12 - 8 = 4$$

Answer to exercise 21 II

3. $[K(\mathbf{x}_1, \mathbf{x}_m)]_m$ is the first line of $\mathbf{K}$, $[2, 0, 2]$ so $\mathcal{K}(\mathbf{x}_1) = [K(\mathbf{x}_1, \mathbf{x}_m)]_m \mathbf{K}^{-1} = [1, 0, 0]$ and $\mathcal{K}(\mathbf{x}_1)\omega(\mathbf{X}) = [1, 0, 1, 0, 0]$
   $[K(\mathbf{x}_2, \mathbf{x}_m)]_m$ is the first line of $\mathbf{K}$, $[0, 2, 2]$ so $\mathcal{K}(\mathbf{x}_2) = [K(\mathbf{x}_2, \mathbf{x}_m)]_m \mathbf{K}^{-1} = [0, 1, 0]$ and $\mathcal{K}(\mathbf{x}_2)\omega(\mathbf{X}) = [0, 1, 0, 0, 1]$
   $[K(\mathbf{x}_3, \mathbf{x}_m)]_m$ is the first line of $\mathbf{K}$, $[2, 2, 5]$ so $\mathcal{K}(\mathbf{x}_3) = [K(\mathbf{x}_2, \mathbf{x}_m)]_m \mathbf{K}^{-1} = [0, 0, 1]$ and $\mathcal{K}(\mathbf{x}_3)\omega(\mathbf{X}) = [1, 1, 1, 1, 1]$
4. Let us consider $\mathbf{x}' = [1, -1]$.
   $$\omega(\mathbf{x}') = [1, -1, 1, -1, 1]$$

Answer to exercise 21 III

5. To see if $\omega(\mathbf{x}) \notin \text{span}(\omega(\mathbf{x}_1), \omega(\mathbf{x}_2), \omega(\mathbf{x}_3))$, we set $\alpha, \beta, \gamma, \delta$ such that
$$\alpha\omega(\mathbf{x}_1) + \beta\omega(\mathbf{x}_2) + \gamma\omega(\mathbf{x}_3) + \delta\omega(\mathbf{x}') = 0$$
and we try to show that they are necessarily equal to 0.
$$\begin{cases} \alpha + \gamma + \delta = 0 \\ \beta + \gamma - \delta = 0 \\ \alpha + \gamma + \delta = 0 \\ \gamma - \delta = 0 \\ \beta + \gamma + \delta = 0 \end{cases}$$
And indeed. When we add samples, we quickly get to span the whole constructed feature space.

Answer to exercise 21 IV

6. $$\mathcal{K}(\mathbf{x}') = [\omega(\mathbf{x}').\omega(\mathbf{X})]_n \mathbf{K}^{-1} = [2,\, 0,\, -1]\mathbf{K}^{-1} = [2,\, 1,\, -1]$$
and $\mathcal{K}(\mathbf{x}')\omega(\mathbf{X}) = [1,\, 0,\, 1,\, -1,\, 0]$ Now we want to show that
$$\mathcal{K}(\mathbf{x}')\omega(\mathbf{X}) \notin \omega(\mathscr{F})$$
If this was wrong then there would exists x_1'', x_2'' such that
$$x_1'' = 1, \quad x_2'' = 0, \quad x_1''^2 = 1, \quad x_1'' x_2'' = -1, \quad x_2''^2 = 0$$
This is not possible.

Kernel trick

Basic idea

$$\mathbf{x}.\mathbf{x}' \text{ is replaced by } \mathcal{K}(\mathbf{x}, \mathbf{x}')$$

- $\mathcal{K}$ is called a kernel.
- We only need to have $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \mathcal{K}(\mathbf{x}', \mathbf{x})$.
- We do not need left or right linearity.

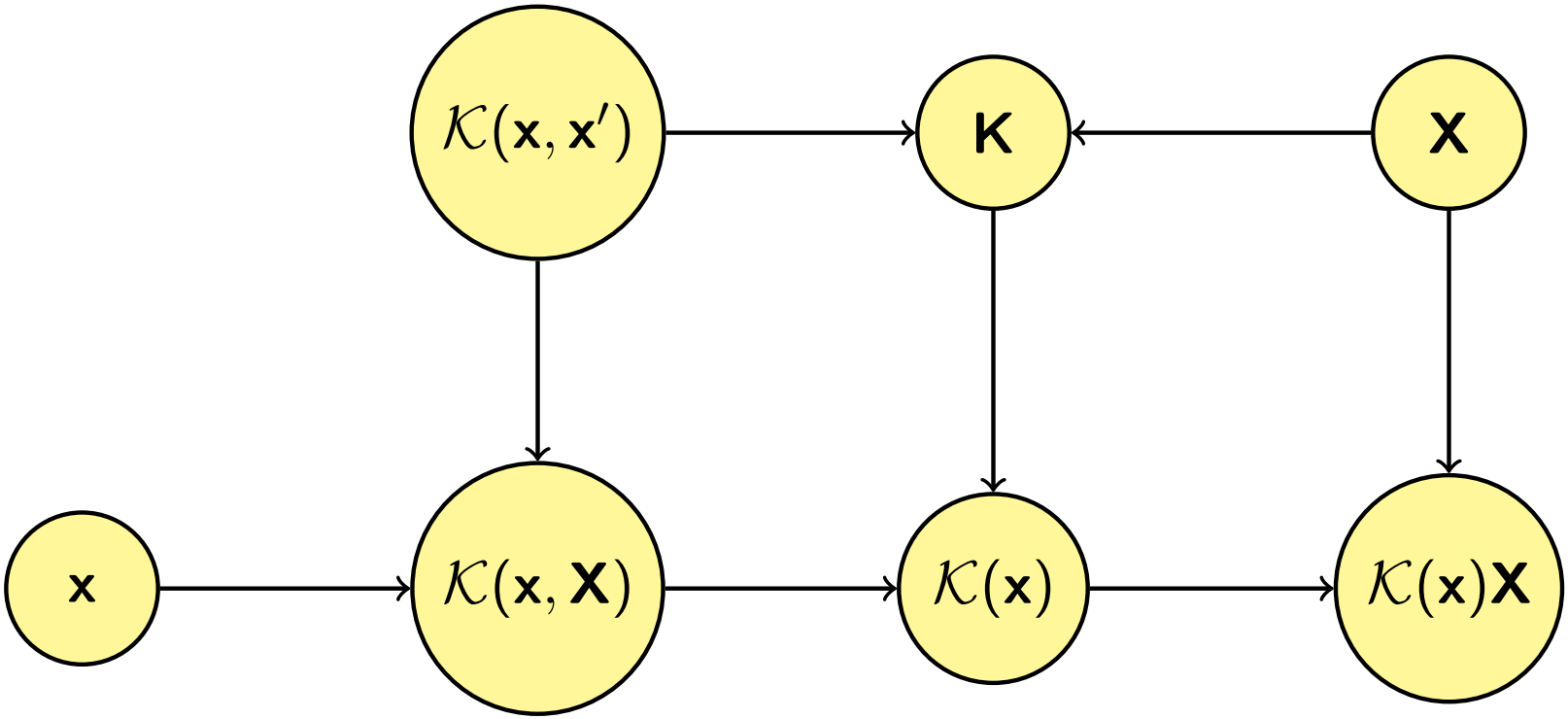
Samples act as a basis

Not an orthogonal basis, but a generally overcomplete basis.

Representing theorem

This theorem states that all samples in the induced feature space can be represented using the data samples using the kernel. It is based on the minimization of a loss function

General scheme



Definitions

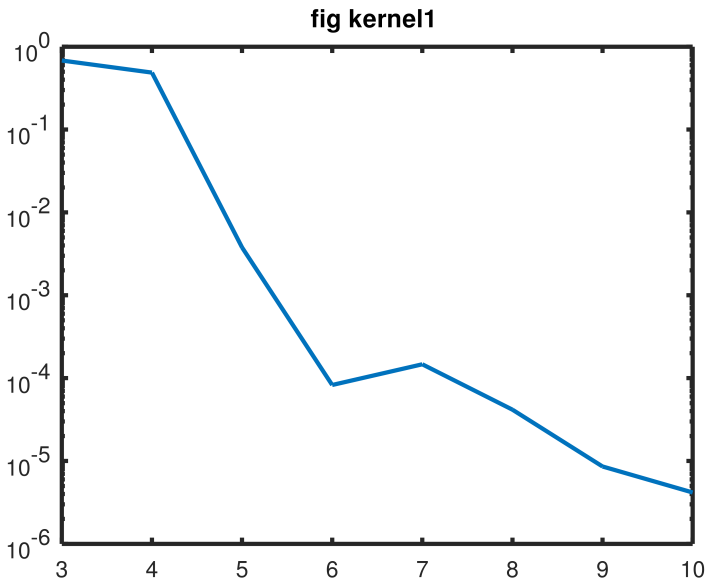
- Kernel matrix
$$\mathbf{K} = [\mathcal{K}(\mathbf{x}_m, \mathbf{x}_n)]_{nm}$$
- Kernel values on the dataset as a row vector
$$[\mathcal{K}(\mathbf{x}, (\mathbf{X})_n)]_n$$
- Mapping in the kernel-induced space
$$\mathcal{K}(\mathbf{x})$$
- Back to the feature space
$$\mathcal{K}(\mathbf{x})\mathbf{X}$$

Nonlinearity remains an issue

This is more adapted to SVM (support vector machine) that uses a dual expression.

Testing the representation theorem

- The X-axis is N
- The Y-axis is $\underset{\mathbf{x}\in\mathscr{F}}{\text{mean}}\left(\left\|\omega\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right)-\mathcal{K}(\mathbf{x})\mathbf{X}\right\|\right)$
- Samples are drawn with $\overset{r}{\mathbf{x}}\sim\mathcal{N}(0,\text{diag}([1\ 1]))$
- The average is computed 10000 experiments.



Exercise 22

Write an algorithm to test the representation theorem on the kernel derived from $\mathbf{x}\mapsto\omega(\mathbf{x})$.

Answer to exercise 22 I

Require: N, I
Ensure: d
1: $d = 0$
2: **for** $i = 1 : I$ **do**
3: Draw the N samples to get $\mathbf{X}$
4: Compute $\omega(\mathbf{X})$
5: Compute $\mathbf{K}$
6: Set $\mathbf{K} := \mathbf{K} + 10^{-5}\mathbf{I}$
7: Compute $\mathbf{K}^{-1}$
8: Draw $\mathbf{x}$ and normalize it.
9: Compute $\mathbf{x}' = [\omega(\mathbf{x}).\omega(\mathbf{X}(1,:))\ldots]\mathbf{K}^{-1}$
10: Update d with $d := d + \|\mathbf{x}'\omega(\mathbf{X}) - \omega(\mathbf{x})\|_{\mathscr{F}}^{\omega}$.
11: $d := \frac{d}{I}$

Conclusion of subsection 3, Kernel trick

- To represent samples in a feature space, it is custom to use an orthonormal basis, with which we have

$$\mathbf{x} = \sum_{n=1}^{N} (\mathbf{e}_n \cdot \mathbf{x}_n)\mathbf{e}_n$$

- Here we have a more general representing technique. Instead of using orthogonality we inverse a matrix.
- And when that matrix is singular we add a diagonal matrix. This is regularization.

Why could this be a problem to add features?

We have seen technique to increase the number of features. We are going to see that this could be an issue.

Content of section 4, Curse of dimensionality, regularization and sparsity

Reasons to do feature extraction?

Numerical complexity

As of now, time is generally not the main issue. However numerical complexity can increase exponentially. We might choose to use the increase numerical complexity for other task.

Hughes phenomenon

This is also called the **curse of dimensionality**.
If when inverting a matrix, you see the following warning, it could be an indication to reduce the dimensionality.

```
warning: matrix singular to machine precision, rcond = 1.5664
warning: called from
```

Example of this phenomenon

The training set contains 10 samples. We use the L_2-solver.

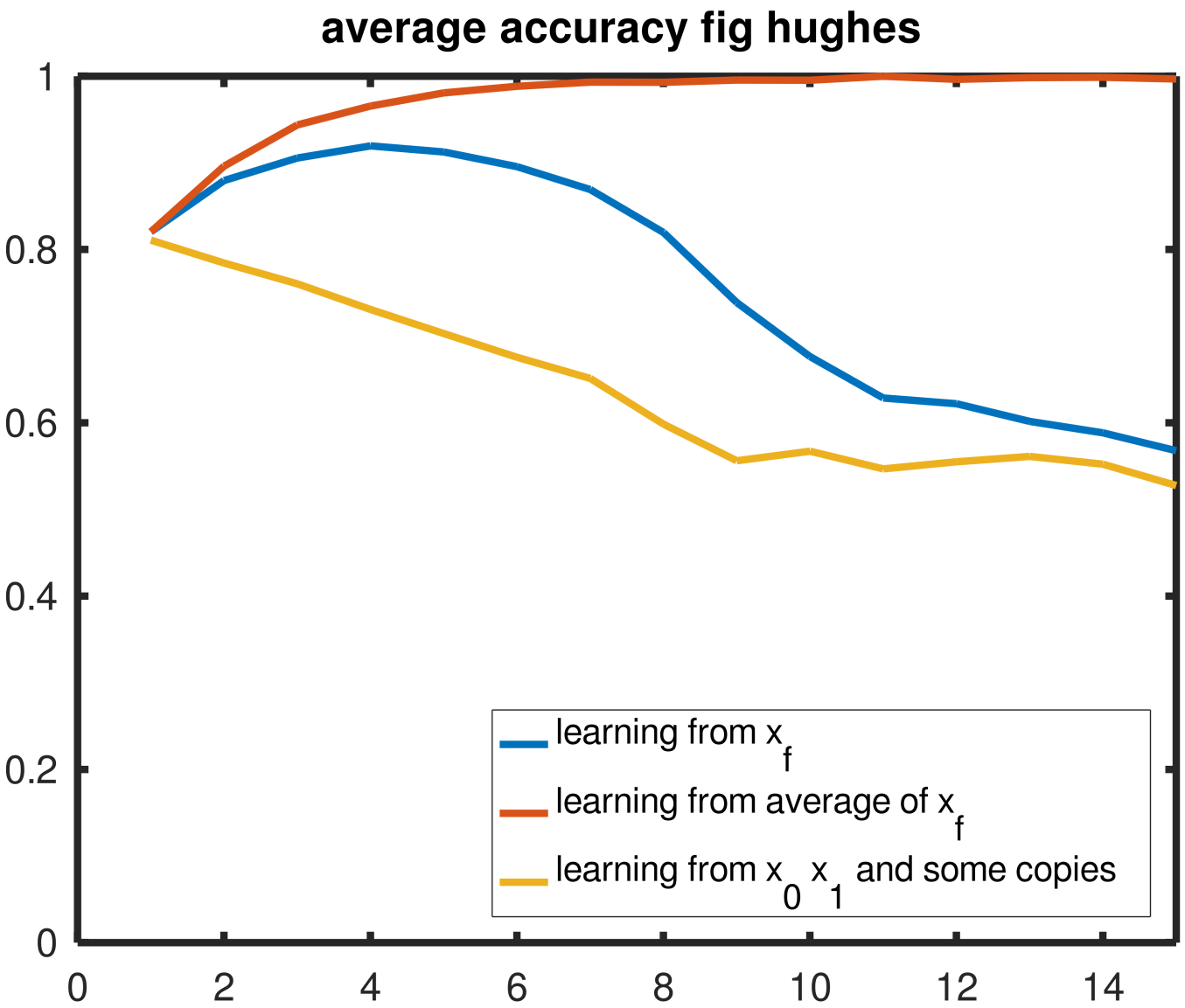
$$\overset{r}{y} \sim \mathcal{U}(\{0,1\}) \text{ and } \overset{r}{\mathbf{x}}|_{y=0} \sim \mathcal{N}(-1,1) \quad \overset{r}{\mathbf{x}}|_{y=1} \sim \mathcal{N}(1,1)$$

Require: F dimension of feature space

Ensure: $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$

1: **for** 500 experiments **do**
2: Draw Y and $\mathbf{X}$
3: Learn $\mathbf{w}_1$ from $\mathbf{X}$ and Y
4: Learn $\mathbf{w}_2$ from $\mathbf{X}1_F^T$ and Y
5: Draw Y_t and $\mathbf{X}_t$
6: Compute $\mathcal{A}_1$ with Y_t and $\mathbf{w}_1$-predictions.
7: Compute $\mathcal{A}_2$ with Y_t and $\mathbf{w}_2$-predictions.
8: Draw x_1 and x_0
9: Draw noisy copies of x_1 and x_0 into $\mathbf{X}_3$, Y_3.
10: Learn $\mathbf{w}_3$ from $\mathbf{X}_3$ and Y_3
11: Compute $\mathcal{A}_3$ with Y_t and $\mathbf{w}_2$-predictions.
12: Average $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$.

Simulations



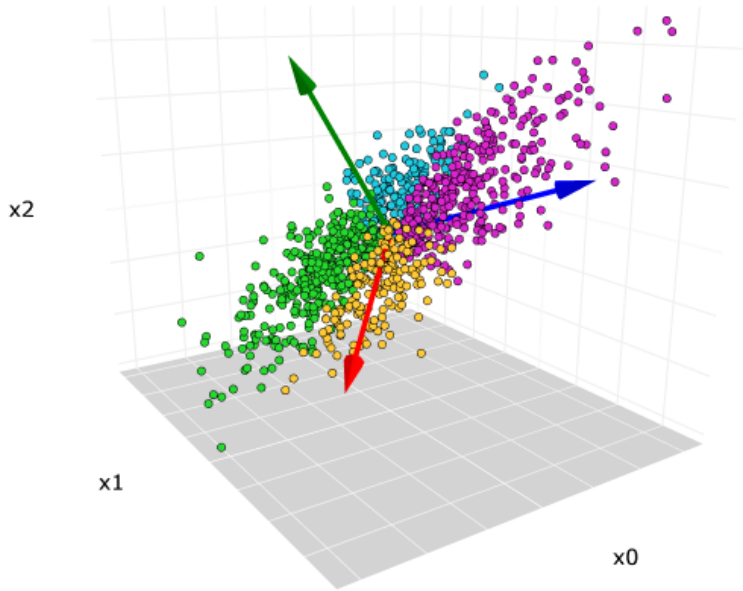
- Experiment
  With feature extraction, we try to find linear combinations of existing features that captures most information.
- Experiment
  With feature selection, we try to keep only the most informative features.

Is a dataset of high dimension?

It is tempting to read this issue from the number of features in a given dataset. However this may not be relevant.

Goal

- Have a reduced number of features. It is also called **dimensionality reduction**.
- **extraction** as opposed to selection, it means that all features changed.



Feature values are changed?

- Stored features values are modified.
- The original feature values can be recovered with the inverse transform (if we do not reduce the number of components).
- Geometric interpretation: same points but different axis and different coordinates.

Conclusion of subsection 4, Curse of dimensionality and feature extraction

To illustrate the need for feature extraction, we made three experiments.

- **x** are drawn with respect to y
- The obtained **x** are replaced by the mean.
- x_1 and x_0 are drawn and the remaining features are copies.

The first experiment shows the need for feature extraction. The third experiment shows the need for feature selection.

A popular feature extraction technique

We will see in detail PCA (principal component analysis), an unsupervised technique.

Content of section 4, Curse of dimensionality, regularization and sparsity

Principal Component Analysis

- Unsupervised technique
- In 2D and 3D, features are rotated.
- New features are ordered by order of importance.
- We may keep only the most important.

PCA: getting the transformation matrix $\mathbf{P}$

Require: $\mathbf{X}$ **centered**

Ensure: $\mathbf{P}$ and $\mathbf{D}$

1: Compute covariance matrix $\mathbf{X}^T\mathbf{X}$
2: Compute the eigenvalue decomposition yielding $\mathbf{V}_1$ and $\mathbf{D}_1$
3: Find a permutation order to have decreasing eigenvalues
4: Apply the permutation order to transform $\mathbf{V}_1$ and $\mathbf{D}_1$ into $\mathbf{P}$ and $\mathbf{D}$

```
[V1,D1]=eig(X'*X);
[~,ind]=sort(D1);
P=V1*eye(size(D))(ind,:);
D=D1*eye(size(D))(ind,:);
```

PCA in a nutshell

Linear algebra

$$\mathbf{X} \quad \rightarrow \quad P, D$$

Matrix computations

$$P \quad \rightarrow \quad \text{PCA}_{\mathcal{P}}$$

$$P, F_1 \quad \rightarrow \quad \text{PCA}_{\mathcal{TP}}, \ \text{PCA}_{\mathcal{T}}$$

$$D, F_1 \quad \rightarrow \quad \mathcal{A}_{\mathcal{T}\text{PCA}}$$

Analysis and synthesis

$$\mathbf{x} \quad \rightarrow \quad \overbrace{\text{PCA}_{\mathcal{P}}(\mathbf{x})}^{\in \mathcal{P}} \quad \rightarrow \quad \mathbf{x}$$

Approximation

$$\mathbf{x} \quad \rightarrow \quad \overbrace{\text{PCA}_{\mathcal{P}}(\mathbf{x})}^{\in \mathcal{P}} \quad \overset{F_1}{\rightarrow} \quad \overbrace{\text{PCA}_{\mathcal{TP}}(\mathbf{x})}^{\in \mathcal{P} \text{ or } \in \mathcal{P}_{F_1}} \quad \rightarrow \quad \text{PCA}_{\mathcal{T}}(\mathbf{x})$$

Accuracy of approximation

$$\frac{\|\mathbf{x} - \ \text{PCA}_{\mathcal{T}}(\mathbf{x})\|}{\|\mathbf{x}\|} \text{ is on average equal to } 1 - \mathcal{A}_{\mathcal{T}\text{PCA}}$$

What we get with PCA I

Analysis

Given $\mathbf{x}$, we transform into

$$\mathrm{PCA}_{\mathcal{P}}(\mathbf{x}) = \mathbf{xP}$$

Components are statistically independent from each others

$$f \neq f' \Rightarrow \sum_{n=1}^{N} (\ \mathrm{PCA}_{\mathcal{P}}(\mathbf{x}))_{nf} (\ \mathrm{PCA}_{\mathcal{P}}(\mathbf{x}))_{nf'} = 0$$

We can construct approximations by truncating the vector.

$$\mathrm{PCA}_{\mathcal{TP}}(\mathbf{x}) = \mathbf{xP}\mathrm{diag}([\overbrace{1 \ldots 1}^{<-F_1->}, 0 \ldots 0])$$

What we get with PCA: II

Synthesis

Given $\text{PCA}_{\mathcal{P}}(\mathbf{x})$, we get $\mathbf{x}$

$$\mathbf{x} = \text{PCA}_{\mathcal{P}}(\mathbf{x})\mathbf{P}^T$$

Given the truncated vector $\text{PCA}_{\mathcal{TP}}(\mathbf{x})$, we get a good approximation of $\mathbf{x}$, denoted $\text{PCA}_{\mathcal{T}}(\mathbf{x})$

$$\text{PCA}_{\mathcal{T}}(\mathbf{x}) = \text{PCA}_{\mathcal{TP}}(\mathbf{x})\mathbf{P}^T$$

We also have an orthogonality property

$$\text{PCA}_{\mathcal{T}}(\mathbf{x}).(\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})) = 0$$

The accuracy of the approximation is

$$\mathcal{A}_{\mathcal{T}\text{PCA}} = 1 - \underset{\mathbf{x}\in\mathbf{X}}{\text{mean}} \frac{\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2}{\|\mathbf{x}\|^2}$$

where $\|\mathbf{x}\|^2 = \mathbf{x}.\mathbf{x} = \mathbf{x}\mathbf{x}^T$

Perhaps in terms of accuracy, it would have made more sense to consider

$$1 - \frac{\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|}{\|\mathbf{x}\|}$$

But then we loose an easy connection with variance.

Accuracy as a function of F_1 and D

- PCA yields a diagonal matrix D

$$D = \text{diag}(\lambda_1 \dots \lambda_F)$$

  with $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_F$
- F_1 is the number of components not canceled in $\mathcal{P}$
- The accuracy is

$$\mathcal{A}_{\mathcal{T}\text{PCA}} = \frac{\sum_{f=1}^{F_1} \lambda_f}{\sum_{f=1}^{F} \lambda_f} = \frac{1}{\text{tr}(D)} \sum_{f=1}^{F_1} \lambda_f$$

Illustrating the notations in a toy example I

Exercise 23

We consider a tiny dataset with

$$\mathbf{x}_1 = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

1. *Compute* $\mathbf{X}$ *and* $\mathbf{X}^T\mathbf{X}$

We assume that using a PCA-algorithm we found $\mathbf{P}$ *and* $\mathbf{D}$

$$\mathbf{P} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } \mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{9} \end{bmatrix}$$

2. *Write the analysis and synthesis equations and check that we have a perfect reconstruction.*

Illustrating the notations in a toy example II

Exercise

3. *Considering that we keep only one component, write the approximation scheme.*
4. *Check the orthogonality property.*
5. *Compute* $\|\mathbf{x}\|^2$, $\|\mathbf{x} - \ PCA_{\mathcal{T}}(\mathbf{x})\|^2$
6. *Compute* $\mathcal{A}_{\mathcal{T}PCA}$
7. *Check the* $\mathbf{X}$*-signification of* $\mathcal{A}_{\mathcal{T}PCA}$

Answer to exercise 23 I

$$\mathbf{x}_1 = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

1.

$$\mathbf{X} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix} = \mathbf{X}^T$$

$$\mathbf{X}^T\mathbf{X} = \frac{1}{9}\begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

Answer to exercise 23 II

2.

$$\mathbf{x} \quad \rightarrow \quad \overbrace{\text{PCA}_{\mathcal{P}}(\mathbf{x})}^{\in \mathcal{P}} \quad \rightarrow \quad \mathbf{x}$$

We denote $\mathbf{e}_1^T, \mathbf{e}_2^T$ the column vectors of P

$$\mathbf{P} = [\mathbf{e}_1^T \mathbf{e}_2^T] \text{ with } \mathbf{e}_1 = \frac{\sqrt{2}}{2}[1 \quad 1], \quad \mathbf{e}_2 = \frac{\sqrt{2}}{2}[1 \quad -1]$$

For the analysis we get

$$\text{PCA}_{\mathcal{P}}(\mathbf{x}) = \mathbf{x}\mathbf{P} = [\mathbf{x}\mathbf{e}_1^T \quad bx\mathbf{e}_2^T] = \left[\frac{\sqrt{2}}{2}(x_1 + x_2) \quad \frac{\sqrt{2}}{2}(x_1 - x_2)\right]$$

Denoting the component of $\text{PCA}_{\mathcal{P}}(\mathbf{x})$ as x_1', x_2', we get for the synthesis

$$\text{PCA}_{\mathcal{P}}(\mathbf{x})\mathbf{P}^T = [x_1' \quad x_2'] \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix} = \left[\frac{\sqrt{2}}{2}(x_1' + x_2') \quad \frac{\sqrt{2}}{2}(x_1' - x_2')\right]$$

Answer to exercise 23 III

To check $\text{PCA}_{\mathcal{P}}(\mathbf{x})\mathbf{P}^T = \mathbf{x}$, we check the first component

$$\frac{\sqrt{2}}{2}(x_1' + x_2') = \frac{\sqrt{2}}{2}\left(\frac{\sqrt{2}}{2}(x_1 + x_2) + \frac{\sqrt{2}}{2}(x_1 - x_2)\right) = x_1$$

then the second component

$$\frac{\sqrt{2}}{2}(x_1' - x_2') = \frac{\sqrt{2}}{2}\left(\frac{\sqrt{2}}{2}(x_1 + x_2) - \frac{\sqrt{2}}{2}(x_1 - x_2)\right) = x_2$$

3.

$$\mathbf{x} \quad \to \quad \overbrace{\text{PCA}_{\mathcal{P}}(\mathbf{x})}^{\in \mathcal{P}} \quad \overset{F_1}{\to} \quad \overbrace{\text{PCA}_{\mathcal{T}\mathcal{P}}(\mathbf{x})}^{\in \mathcal{P} \text{ or } \in \mathcal{P}_{F_1}} \quad \to \quad \text{PCA}_{\mathcal{T}}(\mathbf{x})$$

We have shown previously

$$\text{PCA}_{\mathcal{P}}(\mathbf{x}) = \left[\frac{\sqrt{2}}{2}(x_1 + x_2) \quad \frac{\sqrt{2}}{2}(x_1 - x_2)\right]$$

As we keep only the first component,

$$\text{PCA}_{\mathcal{T}\mathcal{P}}(\mathbf{x}) = \frac{\sqrt{2}}{2}(x_1 + x_2)$$

Answer to exercise 23 IV

After synthesis, we get

$$\text{PCA}_{\mathcal{T}}(\mathbf{x}) = \frac{\sqrt{2}}{2}(x_1 + x_2)\mathbf{e}_1 = \begin{bmatrix} \frac{x_1 + x_2}{2} & \frac{x_1 + x_2}{2} \end{bmatrix}$$

4. The difference between $\mathbf{x}$ and its approximation $\text{PCA}_{\mathcal{T}}(\mathbf{x})$ is

$$\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x}) = \begin{bmatrix} \frac{x_1 - x_2}{2} & \frac{x_2 - x_1}{2} \end{bmatrix}$$

The orthogonality property claims that $(\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})) \cdot \text{PCA}_{\mathcal{T}}(\mathbf{x}) = 0$

$$\begin{bmatrix} \frac{x_1 - x_2}{2} & \frac{x_2 - x_1}{2} \end{bmatrix} \cdot \begin{bmatrix} \frac{x_1 + x_2}{2} & \frac{x_1 + x_2}{2} \end{bmatrix} = 0$$

5. The square norm of $\mathbf{x}$ is

$$\|\mathbf{x}\|^2 = \mathbf{x}.\mathbf{x} = x_1^2 + x_2^2$$

The square norm of $\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})$ is

$$\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2 = \frac{(x_1 - x_2)^2}{2}$$

Answer to exercise 23 V

6. Since $\mathbf{D} = \text{diag}([1 \quad \frac{1}{9}])$,

$$\mathcal{A}_{\mathcal{T}\text{PCA}} = \frac{1}{1 + \frac{1}{9}} = \frac{9}{10}$$

7. The signification of $\mathcal{A}_{\mathcal{T}\text{PCA}}$ for $\mathbf{x}$

$$1 - \frac{\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2}{\|\mathbf{x}\|^2} = 1 - \frac{\frac{(x_1 - x_2)^2}{2}}{x_1^2 + x_2^2} = 1 - \frac{1}{2}\frac{(x_1 - x_2)^2}{x_1^2 + x_2^2}$$

When $\mathbf{x} = \mathbf{x}_1$, we get

$$1 - \frac{1}{2}\frac{(\frac{2}{3} - \frac{1}{3})^2}{(\frac{2}{3})^2 + (\frac{1}{3})^2} = 1 - \frac{1}{10}$$

When $\mathbf{x} = \mathbf{x}_2$, we get

$$1 - \frac{1}{2}\frac{(\frac{1}{3} - \frac{2}{3})^2}{(\frac{1}{3})^2 + (\frac{2}{3})^2} = 1 - \frac{1}{10}$$

Hence

$$\underset{\mathbf{x} \in \mathbf{X}}{\text{mean}} \left(1 - \frac{\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2}{\|\mathbf{x}\|^2}\right) = 1 - \frac{1}{10} = \mathcal{A}_{\mathcal{T}\text{PCA}}$$

Insight into the use of $\mathbf{X}^T\mathbf{X}$

A F-multivariate distribution is defined with a mean μ and a covariance matrix Σ

$$f_{\overset{r}{\mathbf{x}}}(\mathbf{x}) = \frac{1}{(2\pi|\det(\Sigma)|)^{F/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)\Sigma^{-1}(\mathbf{x}-\mu^T}$$

$$\Sigma = E\left[(\overset{r}{\mathbf{x}} - \mu)^T(\overset{r}{\mathbf{x}} - \mu)\right]$$

Notation

$\overset{r}{\mathbf{x}}$ denotes a random row vector.

Variance computations

Exercise 24

We consider two independant Gaussian random variable $\overset{r}{z}_1$ *and* $\overset{r}{z}_2$ *centered and normalised.*

$$\overset{r}{z}_1 \sim \mathcal{N}(0,1) \text{ and } \overset{r}{z}_2 \sim \mathcal{N}(0,1)$$

We define a random vector

$$\overset{r}{\mathbf{x}} = \left[\frac{2}{3}\overset{r}{z}_1 + \frac{1}{3}\overset{r}{z}_2, \quad \frac{1}{3}\overset{r}{z}_1 + \frac{2}{3}\overset{r}{z}_2\right]$$

1. *Compute the covariance matrix using* $\Sigma = E\left[(\overset{r}{\mathbf{x}} - \boldsymbol{\mu})^T(\overset{r}{\mathbf{x}} - \boldsymbol{\mu})\right]$

Answer to exercise 24 I

$$\overset{r}{\mathbf{x}} = \left[\frac{2}{3}\overset{r}{z}_1 + \frac{1}{3}\overset{r}{z}_2, \quad \frac{1}{3}\overset{r}{z}_1 + \frac{2}{3}\overset{r}{z}_2\right]$$

1. Here $\boldsymbol{\mu} = 0$ so the covariance matrix is $E[\mathbf{x}^T\mathbf{x}]$.

$$\mathbf{x}^T\mathbf{x} = \begin{bmatrix} \frac{4}{9}(\overset{r}{z}_1)^2 + \frac{1}{9}(\overset{r}{z}_2)^2 + \frac{4}{9}\overset{r}{z}_1\overset{r}{z}_2 & \frac{2}{9}(\overset{r}{z}_1)^2 + \frac{2}{9}(\overset{r}{z}_2)^2 + \frac{5}{9}\overset{r}{z}_1\overset{r}{z}_2 \\ \frac{2}{9}(\overset{r}{z}_1)^2 + \frac{2}{9}(\overset{r}{z}_2)^2 + \frac{5}{9}\overset{r}{z}_1\overset{r}{z}_2 & \frac{4}{9}(\overset{r}{z}_1)^2 + \frac{1}{9}(\overset{r}{z}_2)^2 + \frac{4}{9}\overset{r}{z}_1\overset{r}{z}_2 \end{bmatrix}$$

Because these are independant Gaussian distributions, we have

$$E\left[(\overset{r}{z}_I)^2\right] = E\left[(\overset{r}{z}_2)^2\right] = 1 \text{ and } E\left[\overset{r}{z}_I\overset{r}{z}_2\right] = 0$$

So we get

$$E\left[\mathbf{x}^T\mathbf{x}\right] = \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix}$$

Contours I

Exercise 25

We consider a centered multivariate normal distribution

$$\overset{r}{\mathbf{x}} \sim \mathcal{N}(0, \Sigma) \text{ and } \Sigma = \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix}$$

We want to find the locus of equal density probability of $\mathbf{x}$.

1. *Show that this locus fullfills*
$$J = \frac{1}{2}\mathbf{x}\Sigma^{-1}\mathbf{x}^T$$
with a probability density of $\frac{9}{2\pi}e^{-J}$
2. *Check that*
$$\Sigma^{-1} = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}$$
3. *Defining* $\mathbf{x}$ *with coordinates:* $\mathbf{x} = [x_1 \ x_2]$, *show that they fullfill*
$$2J = 5x_1^2 - 8x_1x_2 + 5x_2^2$$

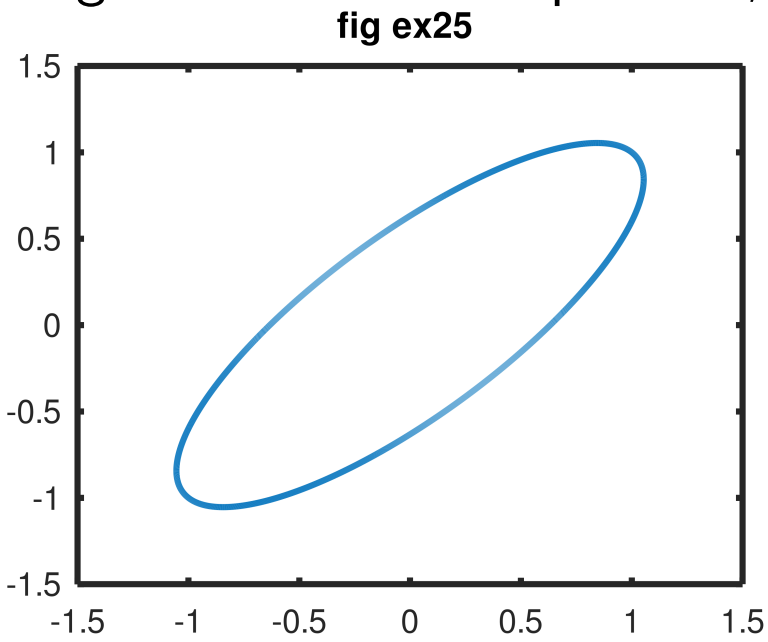
Contours II

Exercise

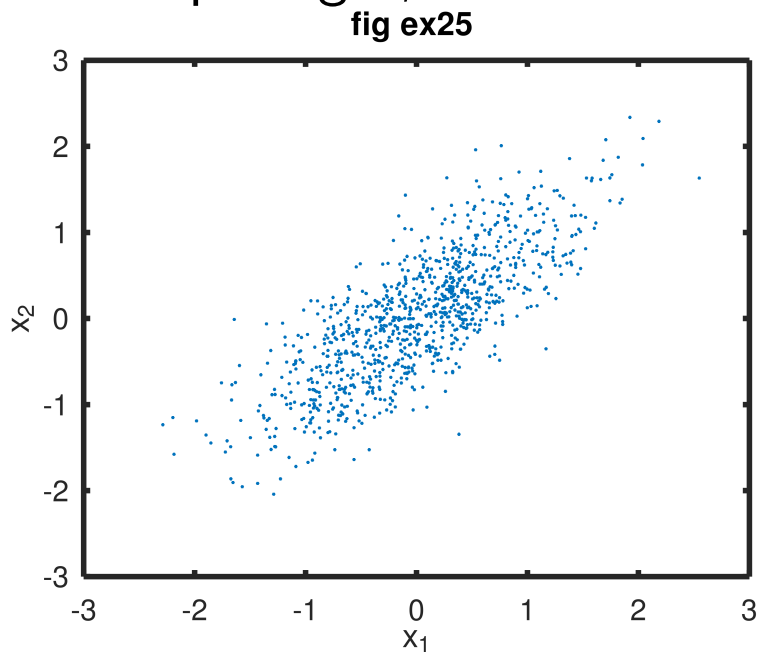
4. *We now use polar coordinates* $x_1 = r\cos(\theta)$ *and* $x_2 = r\sin(\theta)$. *Show that*
$$r(\theta) = \frac{\sqrt{2J}}{\sqrt{5 - 4\sin(2\theta)}}$$
and hence that a parametric description of the contour is
$$\begin{cases} x(\theta) = r(\theta)\cos(\theta) \\ y(theta) = r(\theta)\sin(\theta) \end{cases}$$
5. *Describe the contour and find its closest and farthest points.*
6. *Find a unit vector along the* **farthest** *point's direction. We will see that this is the first eigenvector and hence the first column of the P-matrix.*

Simulations

Using the theoretical equations,



By drawing 1000 points of $\vec{z}_1$, $\vec{z}_2$, and computing $\mathbf{x}$,



Answer to exercise 25 I

$$f_{\mathbf{x}}^{r}(\mathbf{x}) = \frac{1}{2\pi|\det(\Sigma)|} e^{-\frac{1}{2}\mathbf{x}\Sigma^{-1}\mathbf{x}^{T}}$$

1. By defining $J = \frac{1}{2}\mathbf{x}\Sigma^{-1}\mathbf{x}^{T}$, we get

$$f_{\mathbf{x}}^{r}(\mathbf{x}) = \frac{1}{2\pi|\det(\Sigma)|} e^{-J}$$

$$2J = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$2J = \begin{array}{ccccc} 5x_1x_1 & + & -4x_1x_2 & + \\ -4x_2x_1 & + & 5x_2x_2 \end{array}$$

$$2J = 5x_1^2 - 8x_1x_2 + 5x_2^2$$

2. Because $\sin(2\theta) = 2\sin(\theta)\cos(\theta)$, and

$$\det(\Sigma) = \det \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix} = \frac{25-16}{81} = \frac{1}{9}$$

Answer to exercise 25 II

3. $$\Sigma * \Sigma^{-1} = \frac{1}{9}\begin{bmatrix} 25-16 & -20+20 \\ -20+20 & 25-16 \end{bmatrix}$$

4. $$2J = [x_1 \ x_2]\,\Sigma^{-1} = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
$$2J = [5x_1 - 4x_2 \ \ -4x_1 + 5x_2]$$

we get

$$2J = r^2(5 - 4\sin(2\theta))$$

And finally

$$r = \sqrt{\frac{2J}{5 - 4\sin(2\theta)}}$$

Answer to exercise 25 III

5. When $\theta \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, $\theta \mapsto \sin(2\theta)$ is an increasing function, $\theta \mapsto -\sin(2\theta)$ is decreasing and $r = \sqrt{\frac{2J}{5-4\sin(2\theta)}}$ is increasing. The closest point is when $\sin(2\theta)$ is minimal that is $\theta = -\frac{\pi}{4}$ or $\theta = \frac{3\pi}{4}$. The farthest point is when $\sin(2\theta)$ is maximal that is $\theta = \frac{\pi}{4}$ or $\theta = -\frac{3\pi}{4}$. $\theta \mapsto r(\theta)$ ranges between those two extreme points.
6. The farthest point is obtained with $\theta = \frac{\pi}{4}$, that is with $x = \cos(\frac{\pi}{4}) = \frac{\sqrt{2}}{2}$ and $y = \sin(\frac{\pi}{4}) = \frac{\sqrt{2}}{2}$. The corresponding unit vector is $\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$.

Trace and variance

Let $\mathbf{x} \sim \mathcal{N}(0, \Sigma)$

$$\text{var}(\overset{r}{\mathbf{x}}) = E\left[\overset{r}{\mathbf{x}}(\overset{r}{\mathbf{x}})^T\right] = \text{tr}(\Sigma)$$

An experiment

1: **for** $i = 1 : 10^5$ **do**
2: Draw randomly Σ of size 5×5.
3: Rescale Σ so that $\text{tr}(\Sigma) = 1$.
4: Draw $\mathbf{x}$ of size 1×5 following $\mathcal{N}(0, \Sigma)$.
5: Store $\mathbf{x}\mathbf{x}^T$
6: Plot histogram of the stored values

The simulation shows:
$\mathbf{x}\mathbf{x}^T$ is very unlikely to be equal to $\text{tr}(\Sigma)$,
the average of $\mathbf{x}\mathbf{x}^T$ is $\text{tr}(\Sigma)$.

Variance

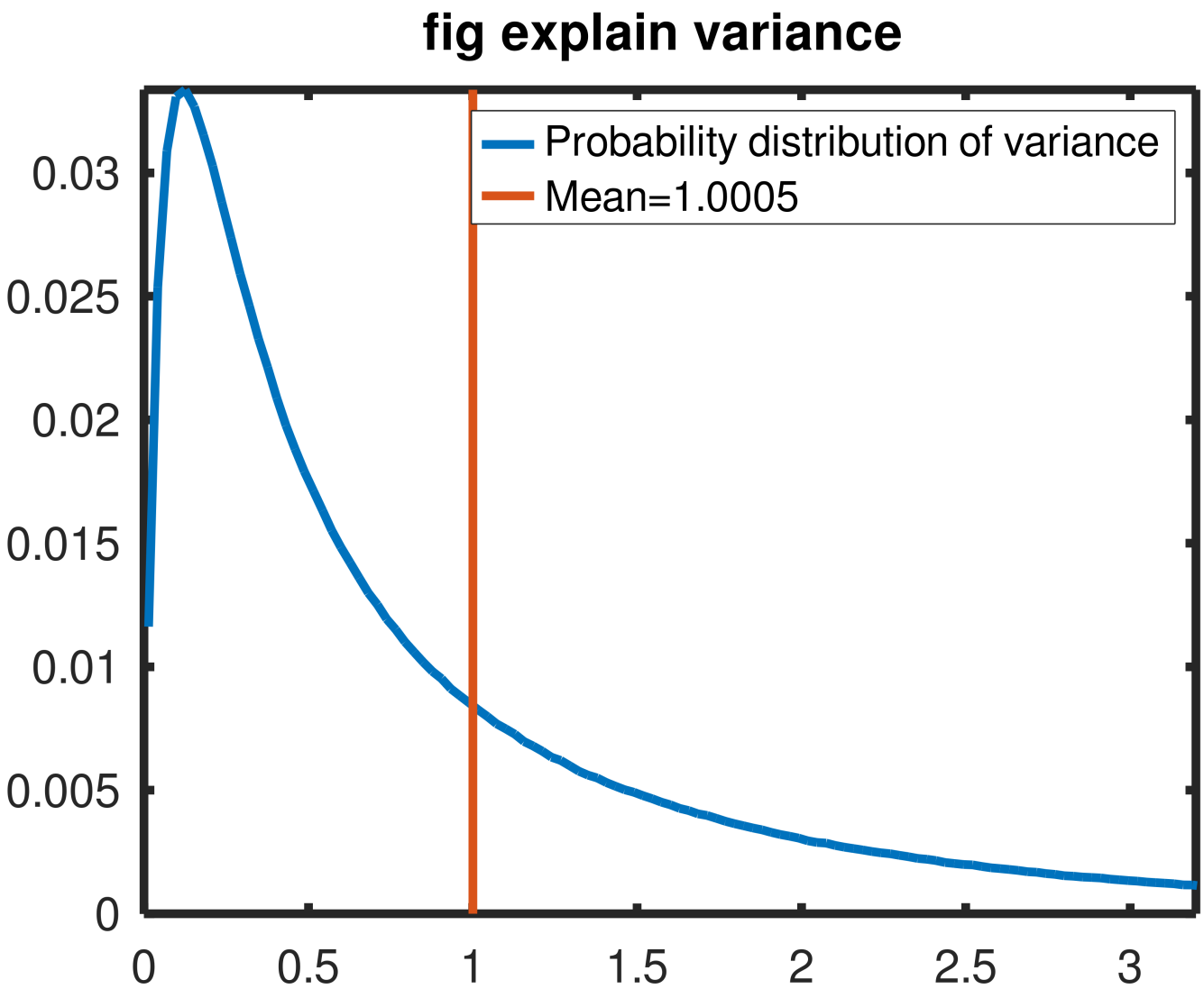


fig explain variance
Probability distribution of variance
Mean=1.0005
0.03
0.025
0.02
0.015
0.01
0.005
0
0
0.5
1
1.5
2
2.5
3

Mean adds to the variance

- In the previous experience $\boldsymbol{\mu} = 0$. If not we have to replace $\mathbf{x}$ with $\mathbf{x} - \boldsymbol{\mu}$.
- The mean's square adds to the variance
$$E[\overset{r}{\mathbf{x}}(\overset{r}{\mathbf{x}})^T] = \text{var}(\overset{r}{\mathbf{x}}) + E[\overset{r}{\mathbf{x}}]E[\overset{r}{\mathbf{x}}]^T = \text{tr}(\Sigma) + \boldsymbol{\mu}\boldsymbol{\mu}^T$$
- In the previous experiment, when we draw $\mathbf{x}$, its mean is non-zero. This non-zero mean is a significant contribution to the measured $\mathbf{x}\mathbf{x}^T$ as $(\mathbf{x} - \text{mean}(\mathbf{x}))(\mathbf{x} - \text{mean}(\mathbf{x}))^T$ would be on average much smaller!

Accuracy of the approximation

Let $\overset{r}{\mathbf{x}} \sim \mathcal{N}(0, \Sigma)$ and $\text{PCA}_{\mathcal{T}}\left(\overset{r}{\mathbf{x}}\right)$ its F_1-component PCA-approximation.

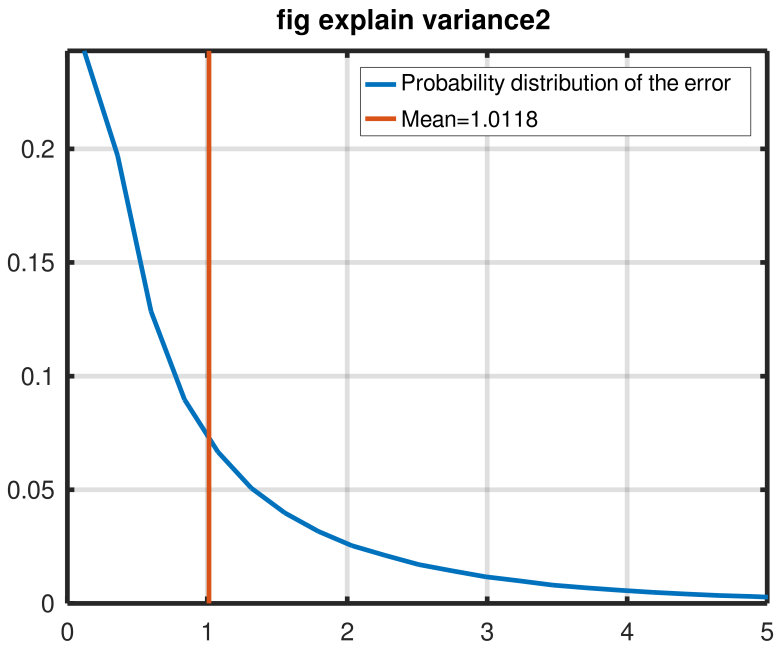
$$E\left[\|\overset{r}{\mathbf{x}} - \text{PCA}_{\mathcal{T}}(\overset{r}{\mathbf{x}})\|^2\right] = (1 - \mathcal{A}_{\mathcal{T},\text{PCA}}(\Sigma, F_1))\,\text{tr}(\Sigma)$$

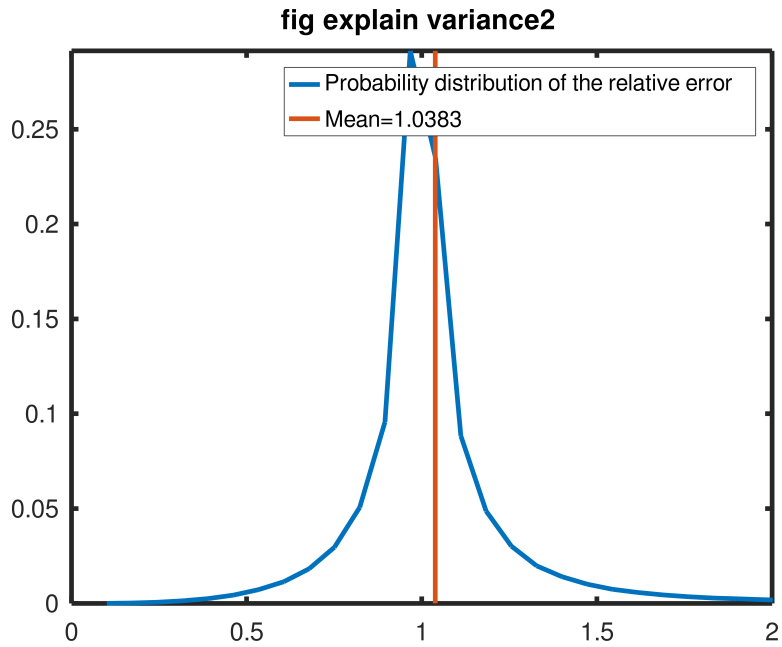
$$E\left[\frac{\|\overset{r}{\mathbf{x}} - \text{PCA}_{\mathcal{T}}(\overset{r}{\mathbf{x}})\|^2}{\|\mathbf{x}\|^2}\right] = 1 - \mathcal{A}_{\mathcal{T},\text{PCA}}(\Sigma, F_1)$$

An experiment

1: **for** $i = 1 : 10^5$ **do**
2: Draw randomly Σ of size 5×5.
3: Rescale Σ so that $\text{tr}(\Sigma) = 1$.
4: With $F_1 = 1$, compute $\mathcal{A}(i) := \mathcal{A}_{\mathcal{T},\text{PCA}}$
5: Draw $\mathbf{x}$ of size 1×5 following $\mathcal{N}(0, \Sigma)$.
6: Compute and store $a(i) := \|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2$
7: Compute and store $b(i) := \left(\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2\right) / \|\mathbf{x}\|^2$
8: Plot histogram of $\frac{a(i)}{1-\mathcal{A}(i)}$ and of $\frac{b(i)}{1-\mathcal{A}(i)}$

Variance 2

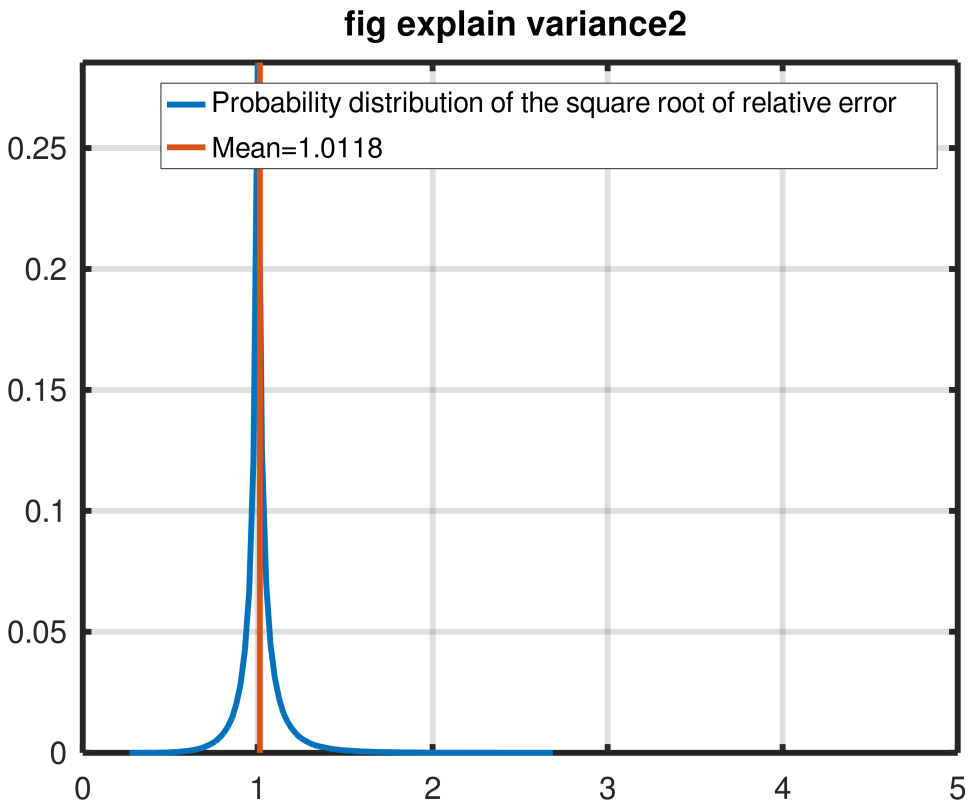




$$\frac{\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2}{1 - \mathcal{A}_{\mathcal{T},\text{PCA}}}$$

$$\frac{\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2}{\|\mathbf{x}\|^2(1 - \mathcal{A}_{\mathcal{T},\text{PCA}})}$$

Probability distribution of the square root of the relative error



$$\frac{\|\mathbf{x} - \mathrm{PCA}_{\mathcal{T}}(\mathbf{x})\|}{\|\mathbf{x}\|\sqrt{1 - \mathcal{A}_{\mathcal{T},\mathrm{PCA}}}}$$

Frobenius norm and variance

- $\| \ \|_{\mathcal{F}}^2$ has a definition using trace.
$$\|\mathbf{X}\|^2 = \text{tr}(\mathbf{X}^T\mathbf{X})$$
- $\| \ \|_{\mathcal{F}}$ is a matrix norm (one among many).
$$\|\mathbf{X}\|_{\mathcal{F}} = \sqrt{\sum_{n,f} x_{nf}^2}$$
- It has a link with the eigenvalue decomposition problem of $\mathbf{X}^T\mathbf{X}$
$$\|\mathbf{X}\|_{\mathcal{F}}^2 = tr(D) = \sum_{f=1}^{F} \lambda_f$$
- It has a link with Σ and variance.

$\mathbf{X}^T\mathbf{X}$ and $\mathbf{x}\mathbf{x}^T$ why?

Here $\mathbf{X}$ is obtained by stacking row vectors $\mathbf{x}_n$

$\mathbf{X}$ is also the concatenation of column vectors X_f.

- $\mathbf{x}\mathbf{x}^T$ is a scalar ($\|\mathbf{x}\|^2$).
- $\mathbf{x}^T\mathbf{x}$ is a $F \times F$ matrix.
- $\frac{1}{\mathbf{x}\mathbf{x}^T}\mathbf{x}^T\mathbf{x}$ is a projector along $\mathbf{x}$.

- $\mathbf{X}^T\mathbf{X}$ is also a $F \times F$ matrix.
$$\mathbf{X}^T\mathbf{X} = \sum_{n=1}^{N} \mathbf{x}_n^T \mathbf{x}_n$$
- $\mathbf{X}^T\mathbf{X}$ is an estimate of the covariance matrix.
$$\mathbf{X}^T\mathbf{X} = [X_f X_{f'}]_{f,f'}$$
- $\mathbf{X}\mathbf{X}^T$ is a $N \times N$ matrix with components $[\mathbf{x}_n \mathbf{x}_{n'}]_{n,n'}$.

$$\|\mathbf{X}\|_{\mathcal{F}}^2 = \mathrm{tr}(\mathbf{X}^T\mathbf{X}) = \mathrm{tr}(\mathbf{X}\mathbf{X}^T)$$

A PCA algorithm I

Projector along axis $\mathbf{e}$

$$\mathcal{P}(\mathbf{x}) = \mathbf{x}\mathbf{e}^T\mathbf{e}$$

When applied to a matrix it renders a matrix whos rows are the projected rows

$$\mathcal{P}(\mathbf{X}) = \mathbf{X}\mathbf{e}^T\mathbf{e}$$

Direction explaining best the variance

We look for $\mathbf{e}$ such that $\mathcal{P}(\mathbf{X})$ is maximal in some sense.

$$\widehat{\mathbf{e}} = \underset{\mathbf{e},\ \|\mathbf{e}\|=1}{\arg\max} \|\mathcal{P}(\mathbf{X})\|_{\mathcal{F}} = \underset{\mathbf{e},\ \|\mathbf{e}\|=1}{\arg\max}\ \mathbf{e}\mathbf{X}^T\mathbf{X}\mathbf{e}^T$$

This could be obtained for instance with `simulated_annealing.m`

```
X=[2/3 1/3; 1/3 2/3];
J=@(e)(-e*X'*X*e')/(e*e');
e=simulated_annealing(J,size(X,2),'silent');
e=(e(:)./sqrt(e(:)'*e(:)))';
```

A PCA algorithm II

Require: $\mathbf{X}$
Ensure: $\mathbf{P}$ and $\mathbf{D}$

1: $\mathbf{X}' := \mathbf{X}$
2: **for** $f = 1 : F$ **do**
3: Compute $\mathbf{e}_f$
4: Project $\mathbf{X}' := \mathbf{X}' - \mathcal{P}(\mathbf{X}')$
5: Update $(\mathbf{X}')^T\mathbf{X}'$
6: $\mathbf{P} := [\mathbf{e}_1^T \dots \mathbf{e}_F^T]^T$
7: $\mathbf{D} := \mathbf{P}^T\mathbf{X}^T\mathbf{X}\mathbf{P}$

Note that with exercise 25 we used this idea to find $\mathbf{e}_1$.

PCA and eigenvalue decomposition

PCA can be regarded as the eigenvalue decomposition of $\mathbf{X}^T\mathbf{X}$.

$$\mathbf{X}^T\mathbf{X} = \mathbf{P}\mathbf{D}\mathbf{P}^T$$

with $\mathbf{P}^T\mathbf{P} = \mathbf{I}_F$ and $\mathbf{D}$ is a $F \times F$ diagonal matrix. This is the idea used in the proposed Matlab/Octave implementation in frame 177.

Eigenvalue decomposition of $\mathbf{X}^T\mathbf{X}$

Because $\mathbf{X}^T\mathbf{X}$ is symmetric, it exists.

- $\mathbf{D} = \text{diag}([\lambda_1 \dots \lambda_F])$ with λ_f as eigen values.
- λ_f are solutions of the polynomial of degree F

$$\det(\mathbf{X}^T\mathbf{X} - \lambda\mathbf{I}_F) = 0$$

- $\mathbf{P} = [\mathbf{e}_1^T \dots \mathbf{e}_F^T]$ with $\mathbf{e}_f$ as eigen vectors.
- $\mathbf{X}^T\mathbf{X}\mathbf{e}_f^T = \lambda_f\mathbf{e}_f^T$ with $\mathbf{e}_f\mathbf{e}_f^T = 1$.

We only need to sort in decreasing order the eigenvalues.

Example of eigenvalued decomposition

Exercise 26

We consider a covariance matrix

$$\Sigma = \frac{1}{9}\begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

We are trying to solve the eigenvalue problem.

1. *Write the second order polynomial yielding the eigenvalues and find them.*
2. *Find the eigenvectors and write the equation.*

Answer to exercise 26 I

1. $$f(\lambda) = \det(\Sigma - \lambda \mathbf{I}_2) = \begin{vmatrix} \frac{5}{9} - \lambda & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} - \lambda \end{vmatrix} = \left(\frac{5}{9} - \lambda\right)^2 - \left(\frac{4}{9}\right)^2$$
$$f(\lambda) = \left(\frac{5}{9} - \lambda - \frac{4}{9}\right)\left(\frac{5}{9} - \lambda + \frac{4}{9}\right)$$
Hence $f(\lambda) = 0 \Leftrightarrow \lambda = 1$ or $\lambda = \frac{1}{9}$
2. We see that if $\mathbf{x} = [1 \quad 1]$,
$$\mathbf{x}\Sigma = [1 \quad 1] = \mathbf{x}$$
So $\mathbf{e}_1 = [1 \quad 1]\frac{\sqrt{2}}{2}$ is the first eigenvector.
We see that if $\mathbf{x} = [1 \quad -1]$,
$$\mathbf{x}\Sigma = [\frac{1}{9} \quad \frac{1}{9}] = \mathbf{x}$$
So $\mathbf{e}_2 = [1 \quad -1]\frac{\sqrt{2}}{2}$ is the second eigenvector.

$$\Sigma\mathbf{P} = \Sigma[\mathbf{e}_1^T \mathbf{e}_2^T] = [\mathbf{e}_1^T \quad \frac{1}{9}\mathbf{e}_2^T] = \mathbf{PD}$$

Definition of SVD

$$\mathbf{X} = \mathbf{U}\Sigma_D\mathbf{V}^T$$

where Σ_D is $N \times F$ and diagonal, $\mathbf{U}\mathbf{U}^T = \mathbf{I}_N$, $\mathbf{V}\mathbf{V}^T = \mathbf{I}_F$

$$\mathbf{X}^T\mathbf{X} = \mathbf{P}\mathbf{D}\mathbf{P}^T = \mathbf{V}\Sigma_D^T\mathbf{U}^T\mathbf{U}\Sigma_D\mathbf{V}^T = \mathbf{V}\Sigma_D^T\Sigma_D\mathbf{V}^T$$

So we have

$$\mathbf{V} = \mathbf{P} \text{ and } (\Sigma_D)_{ff} = \sqrt{(\mathbf{D})_{ff}}$$

Whitening the process

Deterministic

We assume $\mathbf{X} = [\mathbf{x}_1^T \dots \mathbf{x}_N^T]^T$

$$\mathbf{X}^T\mathbf{X} = \mathbf{P}\mathbf{D}\mathbf{P}^T$$

with $\mathbf{P}\mathbf{P}^T = I$ and $\mathbf{D}$ diagonal.

The whitened vector is

$$\mathbf{x} \mapsto \mathbf{z} = \mathbf{x}\mathbf{P}\mathbf{D}^{-1/2}$$

The covariance matrix of $\mathbf{Z} = [\mathbf{z}_1^T \dots \mathbf{z}_N^T]^T$ is

$$\begin{aligned}\mathbf{Z}^T\mathbf{Z} &= \mathbf{D}^{-1/2}\mathbf{P}^T\mathbf{X}^T\mathbf{X}\mathbf{P}\mathbf{D}^{-1} \\ &= \mathbf{D}^{-1/2}\mathbf{P}^T(\mathbf{P}\mathbf{D}\mathbf{P}^T)\mathbf{P}\mathbf{D}^{-1/2} \\ &= \mathbf{I}_F\end{aligned}$$

Statistic

We assume $\mathbf{X} = [\mathbf{x}_1^T \dots \mathbf{x}_N^T]^T$

$$\Sigma_D = \mathbf{P}\mathbf{D}\mathbf{P}^T$$

with $\mathbf{P}\mathbf{P}^T = \mathbf{I}$ and $\mathbf{D}$ diagonal.

The whitened vector is

$$\mathbf{z} = \mathbf{x}\mathbf{P}\mathbf{D}^{-1/2}$$

Components of $\mathbf{z}$, z_f are independent centered normalized Gaussians.

$$\mathbf{x} \mapsto \mathbf{z} = \mathbf{x}\mathbf{P}\mathbf{D}^{-1/2}$$

with

$$\mathbf{X}^T\mathbf{X} = \mathbf{P}\mathbf{D}\mathbf{P}^T$$

We get independent normalized Gaussian random variables

$$\vec[r]{\mathbf{z}}_f \sim \mathcal{N}(0,1)$$

and a white covariance matrix

$$\mathbf{Z}^T\mathbf{Z} = I$$

$$\mathbf{x} \mapsto \mathbf{x}' = \mathbf{x}\mathrm{diag}(\mathbf{X}^T\mathbf{X})^{-1/2}$$

We get unitary random components

$$\forall f, \quad \mathrm{var}(\vec[r]{x'}_f) = 1$$

And unitary column vectors

$$\|X'_f\| = 1$$

The diagonal of the covariance matrix is equal to one.

$$\forall f, \quad \left((\mathbf{X}')^T\mathbf{X}'\right)_{ff} = 1$$

Solving the eigenvalue problem on a toy example

Exercise 27

We consider the same centered multivariate normal distribution as defined in exercise 25.

$$\vec{\mathbf{x}} \sim \mathcal{N}(0, \Sigma) \text{ and } \Sigma = \begin{bmatrix} \frac{5}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{5}{9} \end{bmatrix}$$

We assume that using a PCA-algorithm we found P and D

$$\mathbf{P} = \frac{\sqrt{2}}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } \mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{9} \end{bmatrix}$$

1. *Write the equations of the whitening process transforming $\vec{\mathbf{x}}$ into $\vec{\mathbf{z}}'$.*

We now assume as in exercise 24 that actually $\vec{\mathbf{x}}$ comes from two centered normalized Gaussian random variable $\vec{z}_1$ and $\vec{z}_2$.

$$\vec{x}_1 = \frac{2}{3}\vec{z}_1 + \frac{1}{3}\vec{z}_2 \text{ and } \vec{x}_2 = \frac{1}{3}\vec{z}_1 + \frac{2}{3}\vec{z}_2$$

2. *Check that $\vec{\mathbf{z}}$ is indeed white.*

Answer to exercise 27 I

$$\mathbf{P} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } \mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{9} \end{bmatrix}$$

1. Whitening means that $\mathbf{z}' = \mathbf{x}\mathbf{P}\mathbf{D}^{-1/2}$
$$z_1' = \frac{\sqrt{2}}{2}(x_1 + x_2)$$
$$z_2' = \frac{3\sqrt{2}}{2}(x_1 - x_2)$$
2. We now combine these equations with
$$x_1 = \frac{2}{3}z_1 + \frac{1}{3}z_2 \text{ and } x_2 = \frac{1}{3}z_1 + \frac{2}{3}z_2$$
And we get
$$z_1' = \frac{\sqrt{2}}{2}(z_1 + z_2)$$
$$z_2' = \frac{\sqrt{2}}{2}(z_1 - z_2)$$
which is clearly white as
$$\text{var}(z_1') = \text{var}(z_2') = 1 \text{ and } E[z_1' z_2'] = 0$$

Correlation matrix

We get correlations when we first normalize then compute covariances.

$$\text{corr}\mathbf{X} = \text{cov}\ \text{norm}\mathbf{X} \quad \text{with} \quad \text{norm}\mathbf{X} = \mathbf{X}\text{diag}(\mathbf{X}^T\mathbf{X})^{-1/2}$$

$$\text{and} \quad \text{cov}\mathbf{X} = \mathbf{X}^T\mathbf{X}$$

Its components are estimated with

$$\text{corr}\mathbf{X} = \left[\frac{\sum_{n=1}^{N} x_{nf}x_{nf'}}{\sqrt{\sum_{n=1}^{N} x_{nf}^2}\sqrt{\sum_{n=1}^{N} x_{nf'}^2}}\right]_{ff'} = \frac{(\ \text{cov}\mathbf{X})_{ff'}}{\sqrt{(\ \text{cov}\mathbf{X})_{ff}}\sqrt{(\ \text{cov}\mathbf{X})_{f'f'}}}$$

Its components are between -1 and 1

$$-1 \leq (\ \text{corr}(\mathbf{X}))_{ff'} \leq 1$$

Its diagonal is equal to one.

Here correlation is not concerned with neighboring pixels

It may have to do with neighboring bandwidths.

Example of correlation matrix

Exercise 28

We consider the tiny dataset of exercise 23with

$$\mathbf{x}_1 = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

1. *Compute the correlation matrix.*

Answer to exercise 28 I

$$\mathbf{X} = \frac{1}{3}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

1. $$\text{cov}(\mathbf{X}) = \mathbf{X}^T\mathbf{X} = \frac{1}{9}\begin{bmatrix} 5 & 4 \\ 4 & 5 \end{bmatrix}$$

$$\text{cov}(\mathbf{X}) = \begin{bmatrix} 1 & \frac{4}{5} \\ \frac{4}{5} & 1 \end{bmatrix}$$

because

$$\frac{4}{5} = \frac{\frac{4}{9}}{\sqrt{\frac{5}{9}}\sqrt{\frac{5}{9}}}$$

Conclusion of subsection 5, Principal Component Analysis

PCA is very popular.

- Linear algebra: Eigenvalue decomposition problem and singular value decomposition problem.
- Transformations: analysis/synthesis and whitening
- Uncorrelated and variance explanation
- Trace of the covariance matrix, Frobenius norm and approximation

PCA is unsupervised

The important information may not be obvious. A supervised technique?

Content of section 4, Curse of dimensionality, regularization and sparsity

Transforming PCA into a supervised feature extraction technique

To have zero mean, we consider $\widetilde{y}$ instead of y. We are going to rotate $\mathbf{x}$ into $\mathbf{x}'$ and the question is what for?

Not the cross-covariance matrix

We want to maximize the covariance between $\overset{r}{\mathbf{x}}$ and $\overset{r}{y}$. It is tempting to consider

$$\text{cov}(\overset{r}{\mathbf{x}}\overset{r}{\widetilde{y}}) = [E(\overset{r}{x}_1\overset{r}{\widetilde{y}}) \dots E(\overset{r}{x}_F\overset{r}{\widetilde{y}})]$$

We have seen before in some conditions that $E[\|\overset{r}{\mathbf{x}}\|^2] = \text{tr}(\mathbf{X}^T\mathbf{X})$

PCA with a modification on the covariance matrix

Let $\widetilde{\mathbf{Y}} = \text{diag}(\widetilde{Y})$

$$E[\|\overset{r}{\mathbf{x}}\overset{r}{\widetilde{y}}\|^2] = \text{tr}\left((\widetilde{\mathbf{Y}}\mathbf{X})^T(\widetilde{\mathbf{Y}}\mathbf{X})\right) = \text{tr}\left(\mathbf{X}^T\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{Y}}\mathbf{X}\right)$$

How to find the eigenvectors?

The first eigenvector $\mathbf{e}$ defines a projector on $\mathbf{X}$

$$\mathbf{X}' = \mathbf{X}\mathbf{e}^T\mathbf{e}$$

We get the optimization problem

$$\mathbf{e} = \arg\max_{\mathbf{e}} \ \text{tr}\left((\mathbf{X}')^T\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{Y}}\mathbf{X}'\right) = \arg\max_{\mathbf{e}} \ \mathbf{e}\mathbf{X}^T\widetilde{\mathbf{Y}}^T\widetilde{\mathbf{Y}}\mathbf{X}\mathbf{e}^T$$

subjected to $\|\mathbf{e}\| = 1$.

The new PCA supervised-methodology

We replace $\mathbf{X}^T\mathbf{X}$ with $\mathbf{X}^T\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}\mathbf{X}$.

Conclusion of subsection 6, Supervised feature extraction

1. PCA is the most popular dimensional reduction technique.
2. PCA can be adapted by computing the covariance matrix using $\text{diag}(\widetilde{Y})\mathbf{X}$ instead of $\mathbf{X}$.
3. We have also seen in frame 113 that using LDA we get a new supervised feature.
4. Other techniques make use of labels to select the appropriate number of features.

A different linear classifier

The probabilistic framework yields a different linear classifier. It yields a new feature: the linear hyperplane separating predictions.

Content of section 4, Curse of dimensionality, regularization and sparsity

What it is

About the previous examples of regularization

We had to inverse an ill-conditioned matrix and to achieve this we add $\lambda \mathbf{I}$ with λ could be very small.

Definition of the condition of a matrix

Given a square matrix A we call the condition number of a matrix

$$\kappa(A) = \frac{\max(\sigma(A))}{\min(\sigma(A))}$$

where $\sigma(A)$ is the set diagonal components of D in a singular value decomposion.

$$A = UDV'$$

Smoothing and conditioning?

Exercise 29

1. *What is doing this code?*

```
function fig_cond()
  N=10; F=10; cd=zeros(3); X=randn(N,F);
  for m=1:4
    Xn=X; X=smooth(X')';
    for n=1:3
      Xn=smooth(Xn); cd(m,n)=cond(Xn);
    end
  end
  disp(num2str(round(cd))),
end
function X2=smooth(X1)
  N=size(X1,2); X2=[X1(:,1) (X1(:,1:N-1)+X1(:,2:N))/2];
end
```

Output of an experiment

smoothing $\xrightarrow{\text{along}}$ features

$\downarrow$ smoothing along samples

109	1001	7055
240	1651	9359
2257	13293	59172
17129	96773	395674

Answer to exercise 29 I

Random vectors stacked in $\mathbf{X}$.

$$\overset{r}{\mathbf{x}} \sim \mathcal{N}(0, \text{diag}(1_F))$$

When drawn, the condition number is okay because,

$$(\mathbf{X}^T\mathbf{X})_{mn} \approx N\delta(m = n)$$

The smoothing along the features

$$\mathcal{S}(\mathbf{X}) = \left[\mathbf{X}_{n1}, \frac{\mathbf{X}_{n1} + \mathbf{X}_{n2}}{2}, \dots \frac{\mathbf{X}_{n,F-1} + \mathbf{X}_{nF}}{2}\right]$$

The smoothing along the samples

$$\mathcal{S}(\mathbf{X}^T)^T$$

Border effect

When there are N columns, we can output only $N - 1$ values depending each on two values, if the operations are the same.

Answer to exercise 29 II

Coding the smoothing effect

The computation is operations on a sliding window.

$$(\mathcal{S}(\mathbf{X}))_{nf} = \mathbf{X}_{nf} w_0 + \mathbf{X}_{nf+1} w_1$$

with $w_0 = w_1 = 0.5$.

How can we compute the composition?

$$x'_n = x_n w_0 + x_{n+1} w_1$$
$$x''_n = x'_n w_0 + x'_{n+1} w_1$$

The important property is invariance with respect to a right shift. We see that

$$[w_0 \ w_1] * [w_0 \ w_1] \quad = \quad [w_0^2 \ 2w_0 w_1 \ w_1^2]$$

This is actually the same as polynomial multiplication.

$(w_0 + xw_1)(w_0 + xw_1) = w_0^2 + 2w_0 w_1 x + w_1^2 x^2$

What are the practical consequences?

Because of sensitivity to correlations

- The training set consists of samples drawn randomly in the hyperspectral image. They are not close to each others.
- It is generally a good idea to do dimensionality reduction to reduce correlations among bandwiths.
- However using test samples very close to training samples is an issue.

Modified loss function

L_2-regularization consists in adding

$$\mathcal{L}_{r2}(\mathscr{S}, f^v) = \frac{1}{2}\sum_{n=1}^{N}(b - \mathbf{a}.\mathbf{x}_n - \widetilde{y}_n)^2 + \lambda(b^2 + \|\mathbf{a}\|^2)$$

with $\lambda > 0$ a cost parameter.
This is called the **ridge** OLS.
OLS stands for **Ordinary Least Square**.

Exercise 30

Solve analytically the new optimization problem with the regularized L_2-loss function.

Answer to exercise 30 I

$$2\mathcal{L}_r(\mathscr{S}, f^v) = \left(\overset{\Delta}{\mathbf{X}}\mathbf{w}^T\right)^T \left(\overset{\Delta}{\mathbf{X}}\mathbf{w}^T\right) - \left(\overset{\Delta}{\mathbf{X}}\mathbf{w}^T\right)^T \widetilde{Y}$$

$$-\widetilde{Y}^T \left(\overset{\Delta}{\mathbf{X}}\mathbf{w}^T\right) + \widetilde{Y}^T\widetilde{Y} + \lambda \mathbf{w}\mathbf{w}^T$$

$$2\mathcal{L}(\mathscr{S}, f^v) = \mathbf{w}(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} + \lambda \mathbf{I})\mathbf{w}^T - 2\mathbf{w}\overset{\Delta}{\mathbf{X}}^T \widetilde{Y} + \widetilde{Y}^T\widetilde{Y}$$

And after derivation with respect to $\mathbf{w}$, we get

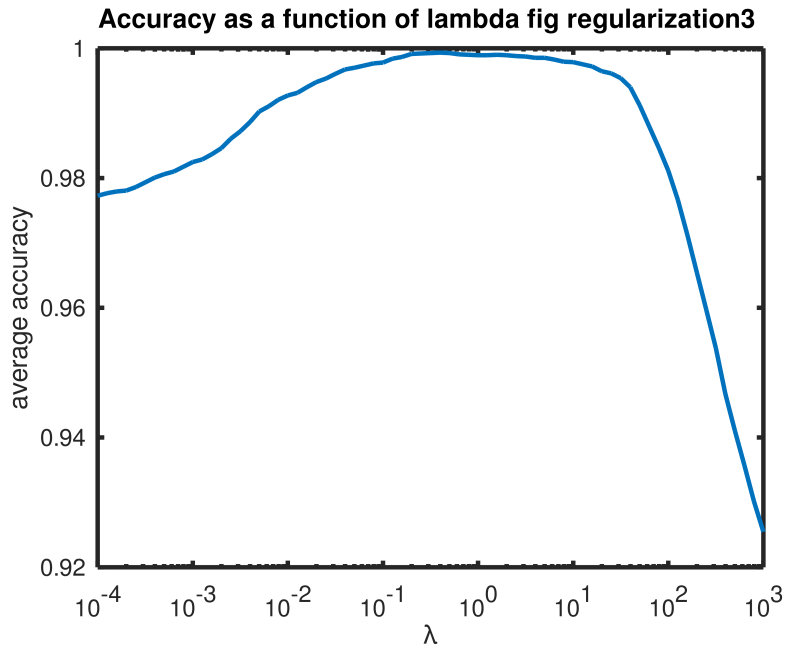
$$\mathbf{w}^T = \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} + \lambda \mathbf{I}\right)^{-1} \overset{\Delta}{\mathbf{X}}^T \widetilde{Y}$$

An experiment showing increased performance with L_2-regularization

Require: λ

Ensure: mean($\mathcal{A}$)

1: **for** 100 experiments **do**
2: Draw a probabilistic problem
3: Draw 10 labeled samples
4: Compute w (ridge OLS)
5: Draw 10 labeled samples
6: Predict 10 labels
7: Measure accuracy
8: Compute average accuracy



$$\overset{r}{\mu}_0, \overset{r}{\mu}_1 \sim \mathcal{N}(0, 4\mathbf{I}_{10}) \text{ and } \overset{r}{\Sigma}_1 \sim \mathcal{U}([0,1]^{10}), \quad \overset{r}{\Sigma}_2 = 0.5(\overset{r}{\Sigma} + (\overset{r}{\Sigma})^T)$$

$$\overset{r}{\mathbf{X}}_{|\overset{r}{Y}=0} \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_2) \text{ and } \overset{r}{\mathbf{X}}_{|\overset{r}{Y}=1} \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_2)$$

Regularization regarded as the choice of an increased prior I

Exercise 31

We consider a regression problem, that is we want to predict **values** *instead of labels. The values are represented by Y. For the sake of simplicity, we consider here only one feature, so the data matrix* $\mathbf{X}$ *is here a column vector X.* $\mathbf{a}$ *is a scalar, a.*

$$Y = aX + \eta$$

a and η are here regarded as a random variable and vector.

$$\overset{r}{a} \sim \mathcal{N}(0, \sigma_a) \text{ and } \overset{r}{\eta} \sim \mathcal{N}(0, \sigma_\eta \mathbf{I}_N)$$

1. *Write the likelihood of Y given X and a.*
2. *Write the posterior probability a given X and Y as a function of the likelihood and a prior.*

Regularization regarded as the choice of an increased prior II

Exercise

3. *Show that $\widehat{a}$ maximizing the posterior probability is defined as*

$$\widehat{a} = \arg\min_{a} (Y - aX)^T (Y - aX) + \frac{\sigma_\eta^2}{\sigma_a^2} a^2$$

Answer to exercise 31 I

1. Denoting the likelihood of Y given X and a

$$f_{Y|X,a}^{r}(X,Y,a) = \frac{1}{\sqrt{2\pi}^{N}|\det(\sigma_\eta^2 \mathbf{I}_N)|^{N/2}} e^{-\frac{1}{2}(Y-aX)^T\left(\sigma_\eta^2 \mathbf{I}_N\right)^{-1}(Y-aX)}$$

$\sigma_\eta^2 \mathbf{I}_N$ is a diagonal matrix whose inverse and determinant are

$$\frac{1}{\sigma_\eta^2}\mathbf{I}_N \text{ and } \sigma_\eta^{2N}$$

This covariance matrix being diagonal we also get the independence among the different components.

$$f_{Y|X,a}^{r}(X,Y,a) = \frac{1}{(2\pi)^{N/2}\sigma_\eta^N} e^{-\frac{1}{2\sigma_\eta^N}(Y-aX)^T(Y-aX)}$$

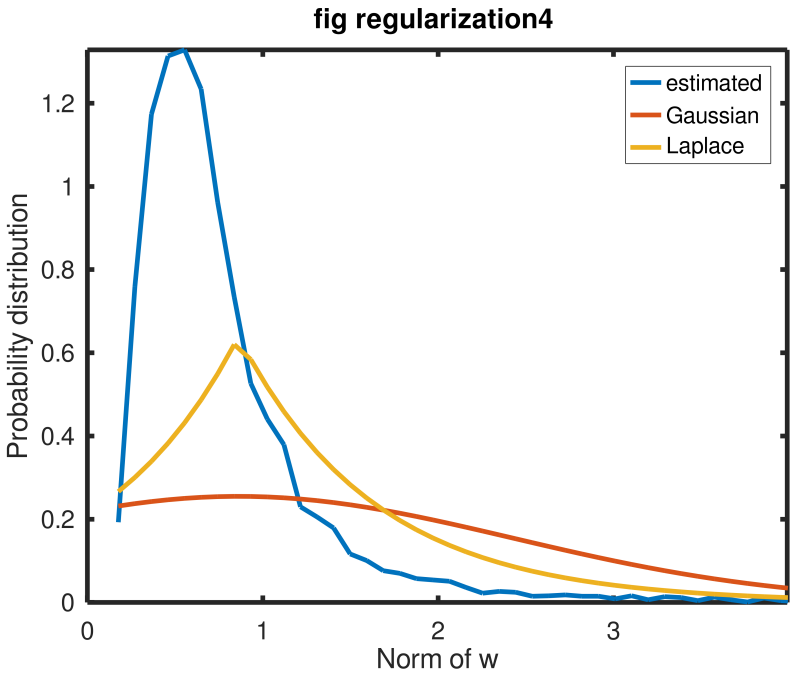
Answer to exercise 31 II

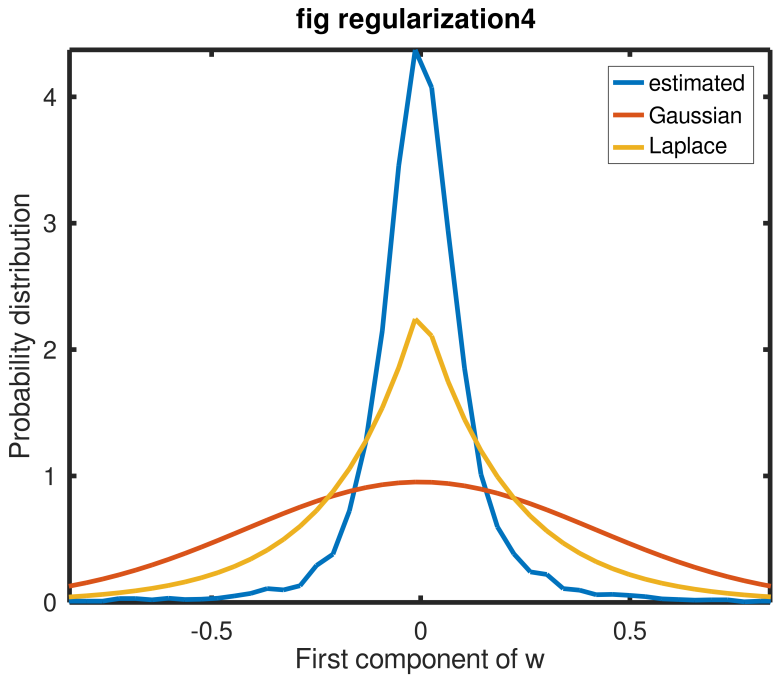
2. The Bayesian formula is sometimes written as
$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\neg A)P(\neg A)}$$
Here this actually means
$$f_{\overset{r}{a}|\overset{r}{Y},X}(X, Y, a) = \frac{f_{\overset{r}{Y}|\overset{r}{a},X}(X, Y, a) f_{\overset{r}{a}}(a)}{\int_{-\infty}^{+\infty} f_{\overset{r}{Y}|\overset{r}{a},X}(X, Y, a) f_{\overset{r}{a}}(a)\, da}$$

3. Because the denominator depends only of X and Y, it is possible to denote its logarithm $-Z(X, Y)$ and hence to get
$$\ln f_{\overset{r}{a}|\overset{r}{Y},X}(X, Y, a) = Z(X, Y) + \ln f_{\overset{r}{Y}|\overset{r}{a},X}(X, Y, a) + \ln f_{\overset{r}{a}}(a)$$
There exists a quantity κ not depending on X and Y such that
$$\begin{aligned}\ln f_{\overset{r}{a}|\overset{r}{Y},X}(X, Y, a) &= Z(X, Y) + \kappa - \frac{1}{2\sigma_\eta^2}(Y - aX)^T(Y - aX) - \frac{1}{2\sigma_a^2}a^2 \\ &= Z(X, Y) + \kappa - \frac{1}{\sigma_\eta^2}\left((Y - aX)^T(Y - aX) + \frac{\sigma_\eta^2}{\sigma_a^2}a^2\right)\end{aligned}$$

$$\mathbf{w}^T = \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} \right)^{-1} \overset{\Delta}{\mathbf{X}}^T Y$$

$$\Rightarrow \mathbf{w}_f \sim \mathcal{L}(0, 0.2) \text{ or } \mathbf{w}_f \sim \mathcal{N}(0, 0.4)$$

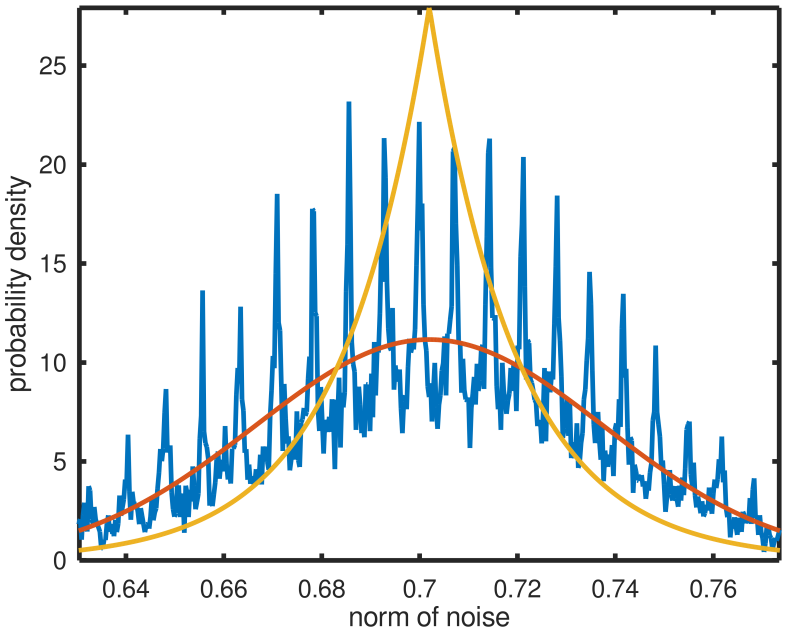


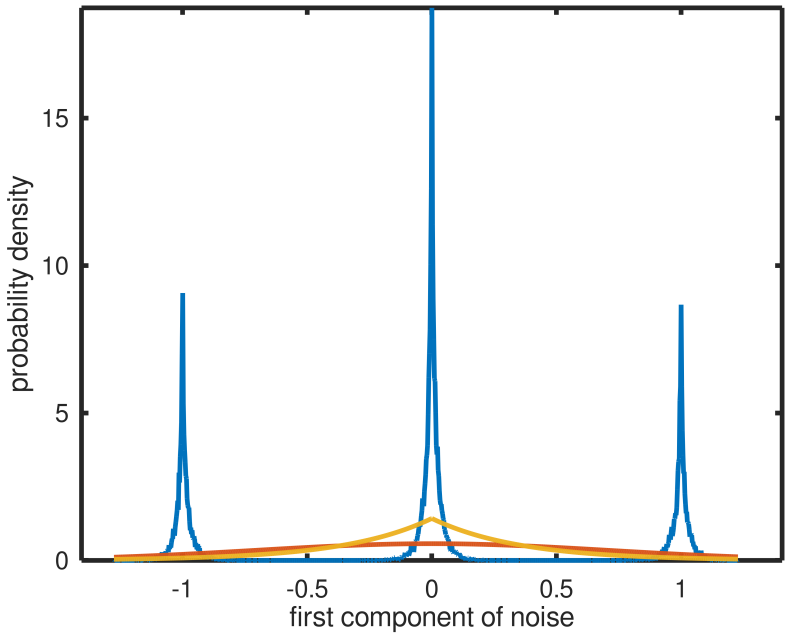


(btw, I did not use here $\widetilde{Y}$)

$$\eta = Y - \mathbf{X}\mathbf{w} \quad \text{with} \quad \mathbf{w}^T = \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1} \overset{\Delta}{\mathbf{X}}^T Y$$

$$\Rightarrow \|\eta\| \sim \mathcal{N}(0, 0.04) \text{ and } \lambda \sim \frac{\sigma_\eta^2}{\sigma_{\mathbf{w}}^2} \approx 10^{-2}$$





Two kinds of regularization for OLS

LASSO

Least absolute shrinkage and selection operator

It is a Laplacian approximation of the parameter prior.

$$\mathcal{L}_{r1}(\mathscr{S}, f^v) = \frac{1}{2}\sum_{n=1}^{N}(b - \mathbf{a}.\mathbf{x}_n - \widetilde{y}_n)^2 + \lambda(|b| + \|\mathbf{a}\|)$$

Ridge OLS

It is a Gaussian approximation of the parameter prior. This regularization is also called the Tikhonov regularization.

$$\mathcal{L}_{r2}(\mathscr{S}, f^v) = \frac{1}{2}\sum_{n=1}^{N}(b - \mathbf{a}.\mathbf{x}_n - \widetilde{y}_n)^2 + \lambda(b^2 + \|\mathbf{a}\|^2)$$

Conclusion of subsection 7, Regularization

1. We first saw a practical classification ill-posed.
2. In the example it results from correlated samples.
3. In an image, training sets and training sets are generally drawn from randomly sampled pixels to avoid such correlations. But practically, this could be an issue for a given application.
4. A Bayesian interpretation of this regularization is given.
5. On an experimental example, it yields two regularization techniques Ridge and LASSO.

Feature selection technique

These regularization techniques yield two feature selection technique.

Content of section 4, Curse of dimensionality, regularization and sparsity

Exercise 32

We consider again exercise 7 and the proposed solution in exercise 9 where

$$\overset{\Delta}{\mathbf{X}} = [\mathbf{X}1], \quad \mathbf{w} = [-\mathbf{a}\ b] \text{ and } \mathbf{w}^T = \left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1} \overset{\Delta}{\mathbf{X}}^T \widetilde{Y}$$

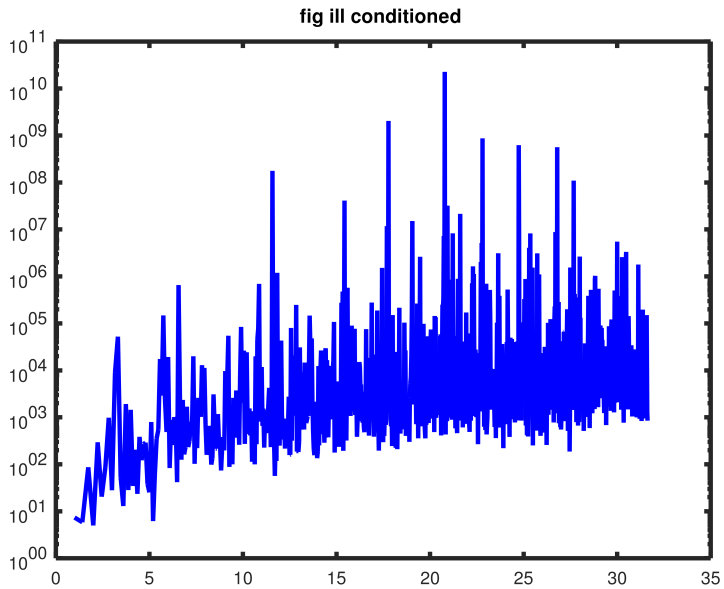
with

$$f_{\mathbf{a},b}(\mathbf{x}) = \delta(\mathbf{a}.\mathbf{x} \leq b)$$

1. *Let us suppose that the first component of all samples in $\mathscr{S}_2$ is constant, why would this be a problem in these equations. Suggest an experiment studying this question.*
2. *What should we think of this situation?*
3. *What could we do?*

Answer to exercise 32 I

1. When first components of all samples have roughly the same value, the first column and the last column of $\overset{\Delta}{\mathbf{X}}$ are proportional and the matrix $\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}$ becomes more and more ill-conditioned.



In this experiment, the first column of $\overset{\Delta}{\mathbf{X}}$ is replaced with ones added to a random number drawn from a centered Gaussian distribution with σ as standard deviation. Each point in this graph indicates vertically the maximum value of the $\left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}}\right)^{-1}$ and horizontally $\frac{1}{\sigma}$.

Require: σ

Answer to exercise 32 II

Ensure: c value of the greatest component

1: Define $\mathbf{X}$ and $\overset{\Delta}{\mathbf{X}}$
2: Draw 3 random values from a Gaussian distribution with mean 1 and standard deviation σ.
3: Replace in $\overset{\Delta}{\mathbf{X}}$ the first column with these values.
4: Compute $\left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} \right)^{-1}$.
5: Let c be the greatest value of $\left(\overset{\Delta}{\mathbf{X}}^T \overset{\Delta}{\mathbf{X}} \right)^{-1}$.

Answer to exercise 32 III

2. If first components of all samples have exactly the same value, say 2, then

$$\delta(b - [a_1, a_2].\mathbf{x} \geq 0) = \delta(0 - [a_1 - \frac{b}{2}, a_2].\mathbf{x} \geq 0)$$

$$= \delta(b - 2a_1 - [0, a_2].\mathbf{x} \geq 0)$$

This identity adds to the general property when $b > 0$,
$\delta(b - [a_1, a_2].\mathbf{x} \geq 0) = \delta(1 - [\frac{a_1}{b}, \frac{a_2}{b}].\mathbf{x} \geq 0)$

3. To cope with this problem, we can just remove this non-informative first component. This is feature selection. (Other ideas could be used too).

Goal in selecting features

Given a dataset (and some information), we would like to select a subset of features.

What for?

Less features decreases the numerical complexity and we may get increased accuracy for a given algorithm. This could be a way to test the efficiency of selecting features.
Another important reason is to yield more understandable predictive models.

Why wouldn't we prefer feature selection rather than feature extraction

To get a more understandable model.
Features could be independent and feature extraction introduces dependency.

Image published by Thenkabail in 2013

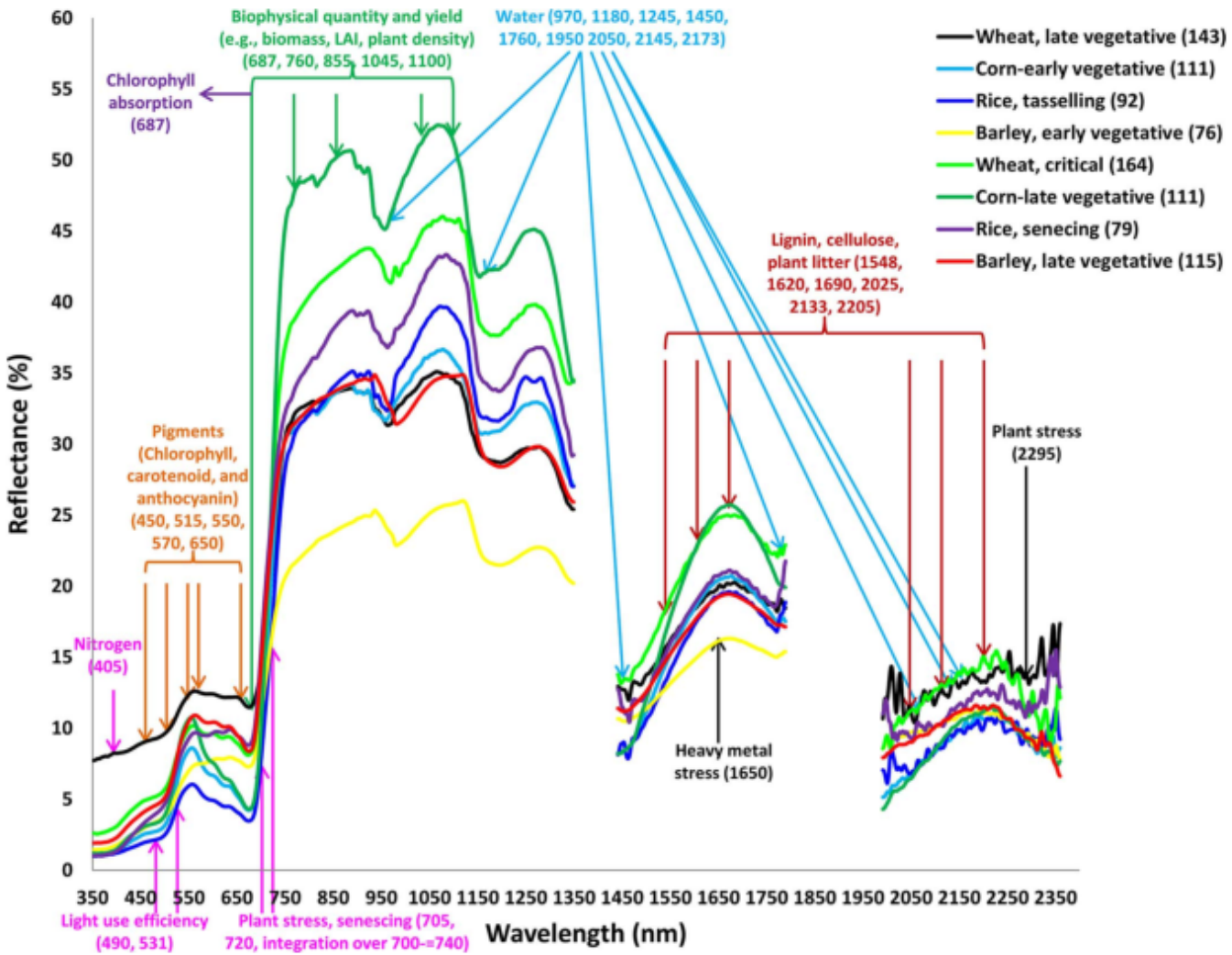


Fig. 8. Optimal hyperspectral narrowbands (HNBs). Current state of knowledge on hyperspectral narrowbands (HNBs) for agricultural and vegetation studies (inferred from [8]). The whole spectral analysis (WSA) using contiguous bands allow for accurate retrieval of plant biophysical and biochemical quantities using methods like continuum removal. In contrast, studies on wide array of biophysical and biochemical variables, species types, crop types have established: (a) optimal HNBs band centers and band widths for vegetation/crop characterization, (b) targeted HVIs for specific modeling, mapping, and classifying vegetation/crop types or species and parameters such as biomass, LAI, plant water, plant stress, nitrogen, lignin, and pigments, and (c) redundant bands, leading to overcoming the Hughes Phenomenon. These studies support hyperspectral data characterization and applications from missions such as Hyperspectral Infrared Imager (HyspIRI) and Advanced Responsive Tactically Effective Military Imaging Spectrometer (ARTEMIS). Note: sample sizes shown within brackets of the figure legend refer to data used in this study.

Image published by Thenkabail in 2013

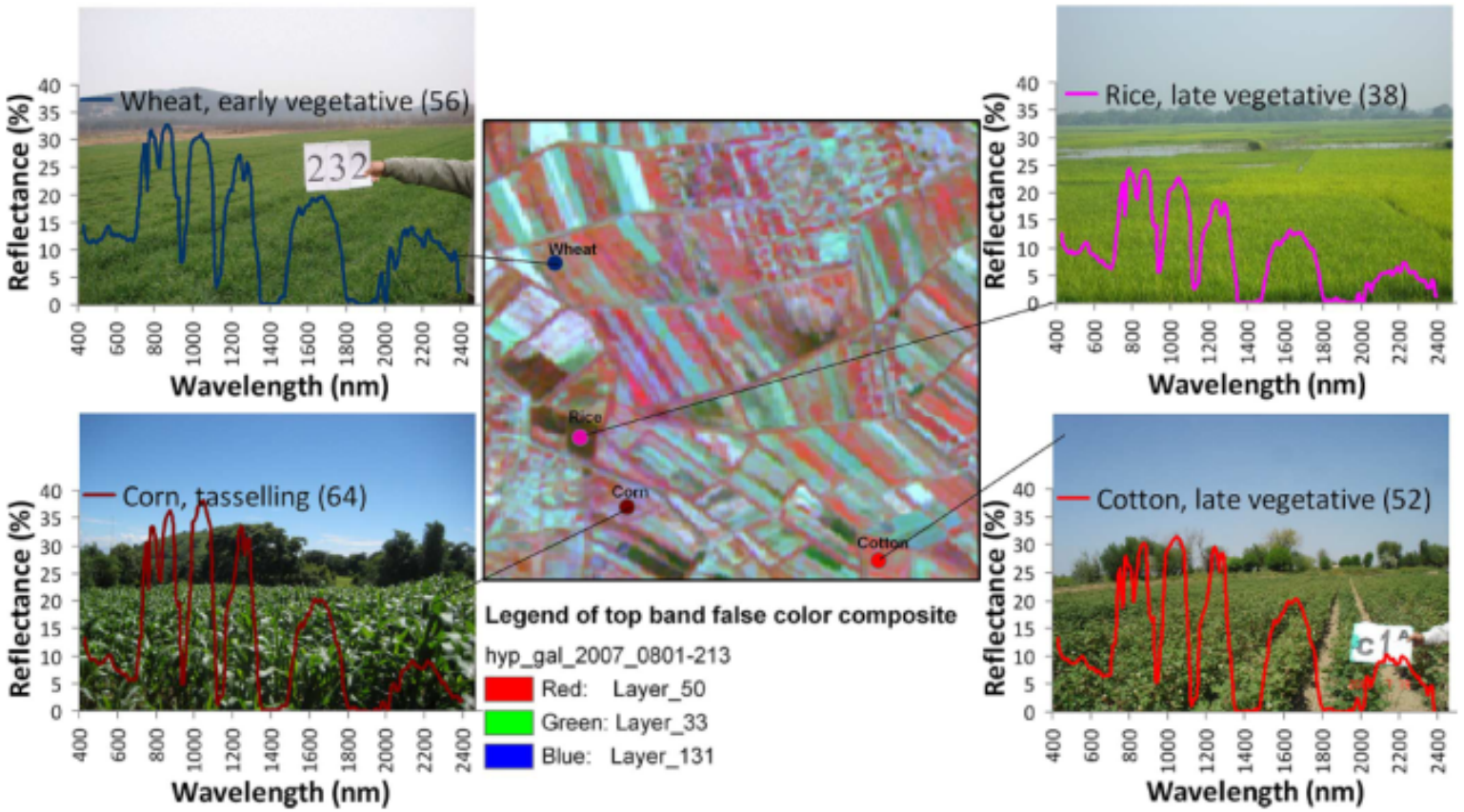


Fig. 3. Hyperion data of crops illustrated for typical growth stages in the Uzbekistan study area. The Hyperion data cube shown here is from a small portion of one of the two Hyperion images. The Hyperion spectra of crops are gathered from different farm fields in the two images and their average spectra illustrated here along with the sample sizes indicated within the bracket. The field data was collected within two days of the image acquisition.

General methodology

There are many feature subsets

$$F = 10, F_1 = 5 \quad \Rightarrow \quad \begin{pmatrix} 10 \\ 5 \end{pmatrix} = 252$$

```
prod(1:10)/prod(1:5)/prod(1:5)
```

- Starting point
  $F_{it=1} = F$ (Backward selection, more popular)
  or $F = 0$ (Forward selection)
- Which feature to select
- Stopping criteria
  Use of validation set.

Require: $\mathscr{F}$
Ensure: $\mathscr{F}'$

1: **repeat**
2: Apply a 1-feature selection technique.
3: Update $\mathscr{F}$
4: **until** Stopping criteria

Two 1-feature selection techniques

Assuming we have decided to remove a feature, which one are we choosing?

The less decrease in accuracy

Ridge:

$$\widehat{f} = \arg\min_{f} |w_f|$$

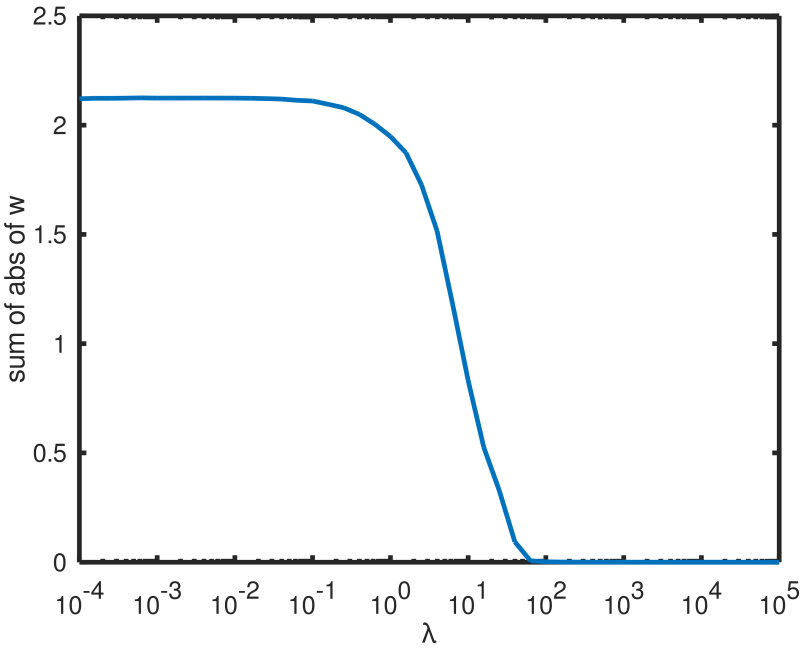
for a given λ.

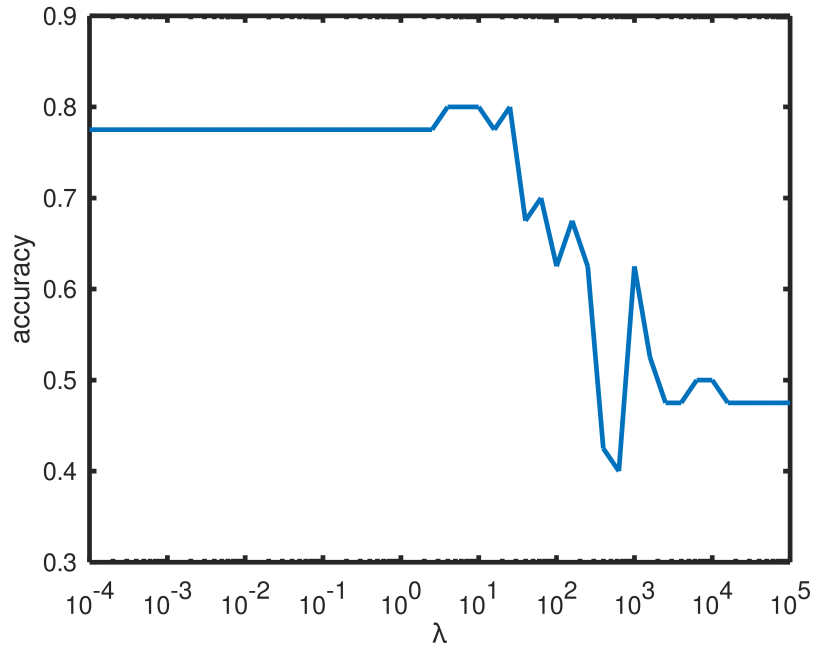
LASSO:

$$\widehat{\lambda} = \arg\min_{\lambda} \{\lambda | \exists f | \widehat{w}_f = 0\}$$

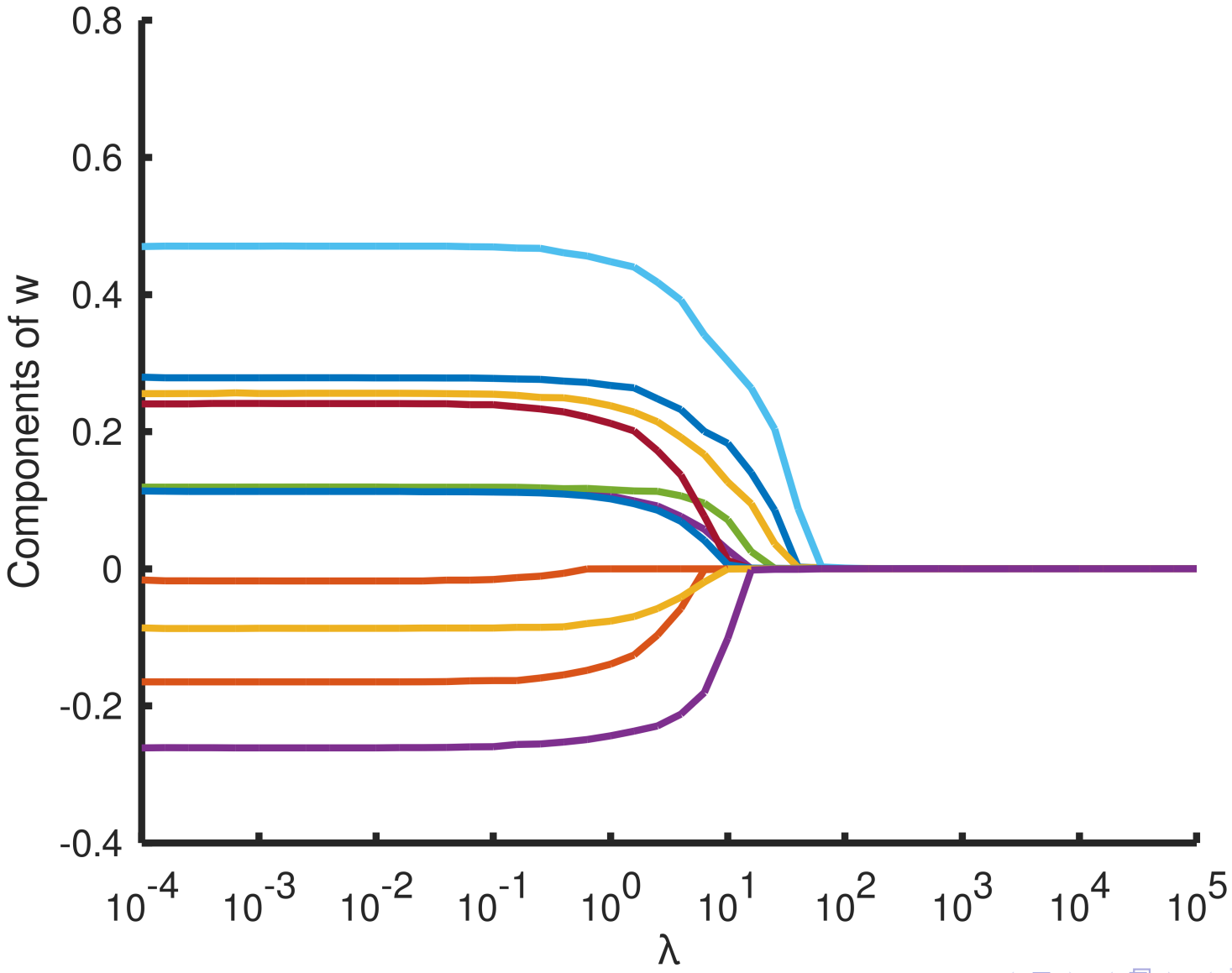
Lasso experiment

$$\mu_0 = 0 \; \mu_1 = [1, 0.9 \ldots 0.1], \; \Sigma = \mathbf{I}_{10}$$





L1 minimization ⇒ features are cancelled



Conclusion of subsection 8, Feature selection

1. Classifying is not only a question of having the best accuracy. Explaining what happens if interesting too.
2. And for hyperspectral images, there is a literature and some specific indexes (NDVI) and many other vegetation indexes.
3. We have discussed the backward and forward feature selection in combination with Ridge regression.
4. We have seen the LASSO feature selection technique.

Spatial context

How these techniques can be applied in a more general context.

Table of Contents

Content of section 5, Spatial context

Different goals

Xie 2015

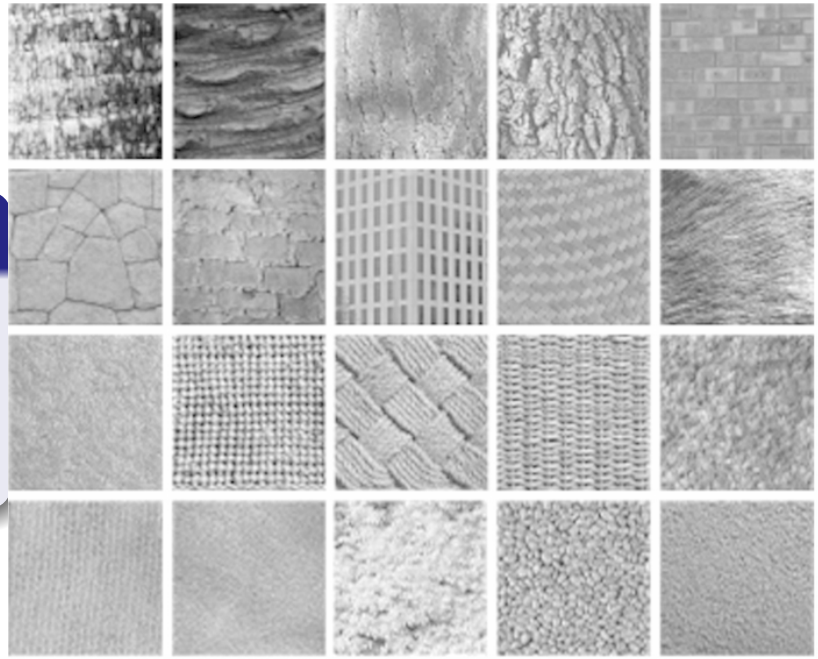
1. Texture (=preprocessing)
2. Measuring the noise (=preprocessing)
3. Prior on the classification map (=post-processing)
4. Mixture of end-vectors
5. Use of Digital Elevation Map (DEM)

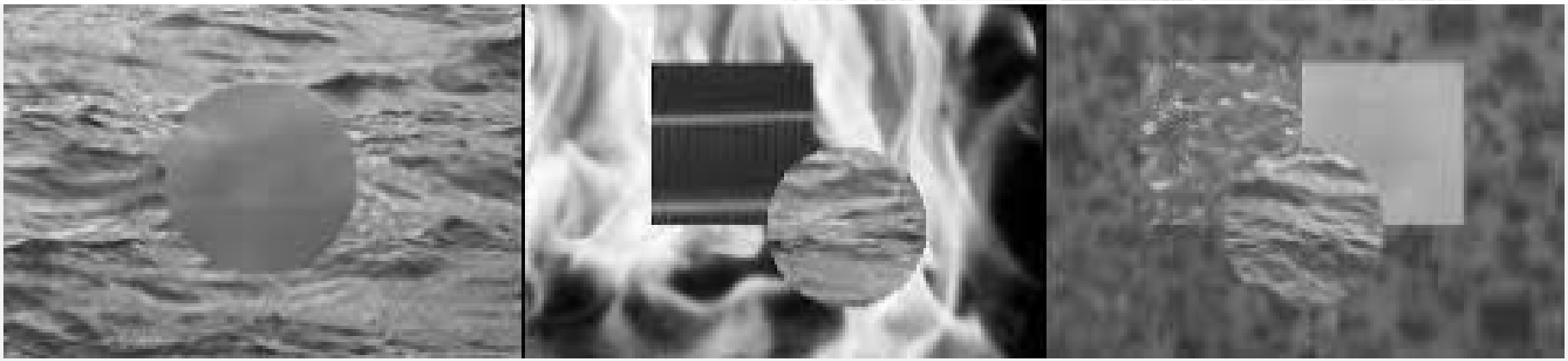
Content of section 5, Spatial context

Rich literature from image processing

What is a texture?

There is no absolute definition. It rather means that we understand the content as a texture.





No perfect tool
How to group the texture descriptors?

Is the technique sensitive to

- a global increase in intensity?
- an image rotation?
- a rescaling of the image?
- a quantification of the image?

Is the technique equivalent to?

- Nonlinear processing, filtering and nonlinear processing?
- Histogram and a diversity index on the histogram?

Proposed techniques I

Let $\mathscr{V}_{mn}$ be the neighborhood of m, n and $\mathscr{V}'_{mn}$ the same neighborhood without the last column.

$$\mathscr{V}_{mn} = \{m', n' | \, max(|m' - m|, |n' - n|) \leq 2\}$$

1. Horizontal filter
$$f'_{mn} = \frac{1}{5} \sum_{m'=m-2}^{m+2} f_{m'n}$$
2. Variance
$$f'_{mn} = \sum_{\mathscr{V}_{mn}} (f_{m'n'} - \mu_{mn})^2$$
$$\text{with } \mu_{mn} = \frac{1}{25} \sum_{\mathscr{V}_{mn}} f_{m'n'}$$
3. Diversity index
$f'_{mn} = \sum_{g} h(g)^2$ where $h(g)$ is the estimated probability distribution

Proposed techniques II

4. Correlation

$$f'_{mn} = \frac{\sum_{\mathcal{V}'_{mn}} f_{m'n'} f_{m'-1n'}}{\sqrt{\sum_{\mathcal{V}'_{mn}} f^2_{m'n'}} \sqrt{\sum_{\mathcal{V}_{mn}} f^2_{m'n'}}}$$

5. Mean

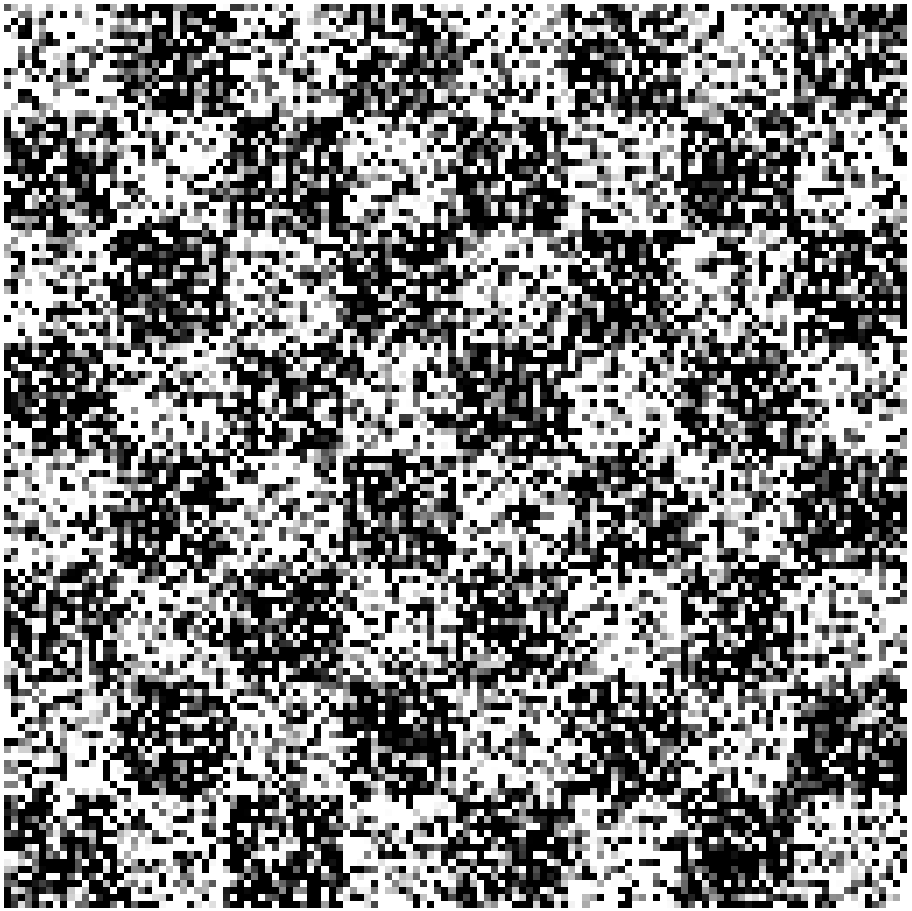
$$f'_{mn} = \frac{1}{25} \sum_{\mathcal{V}_{mn}} f_{m'n'}$$

Exercise 33

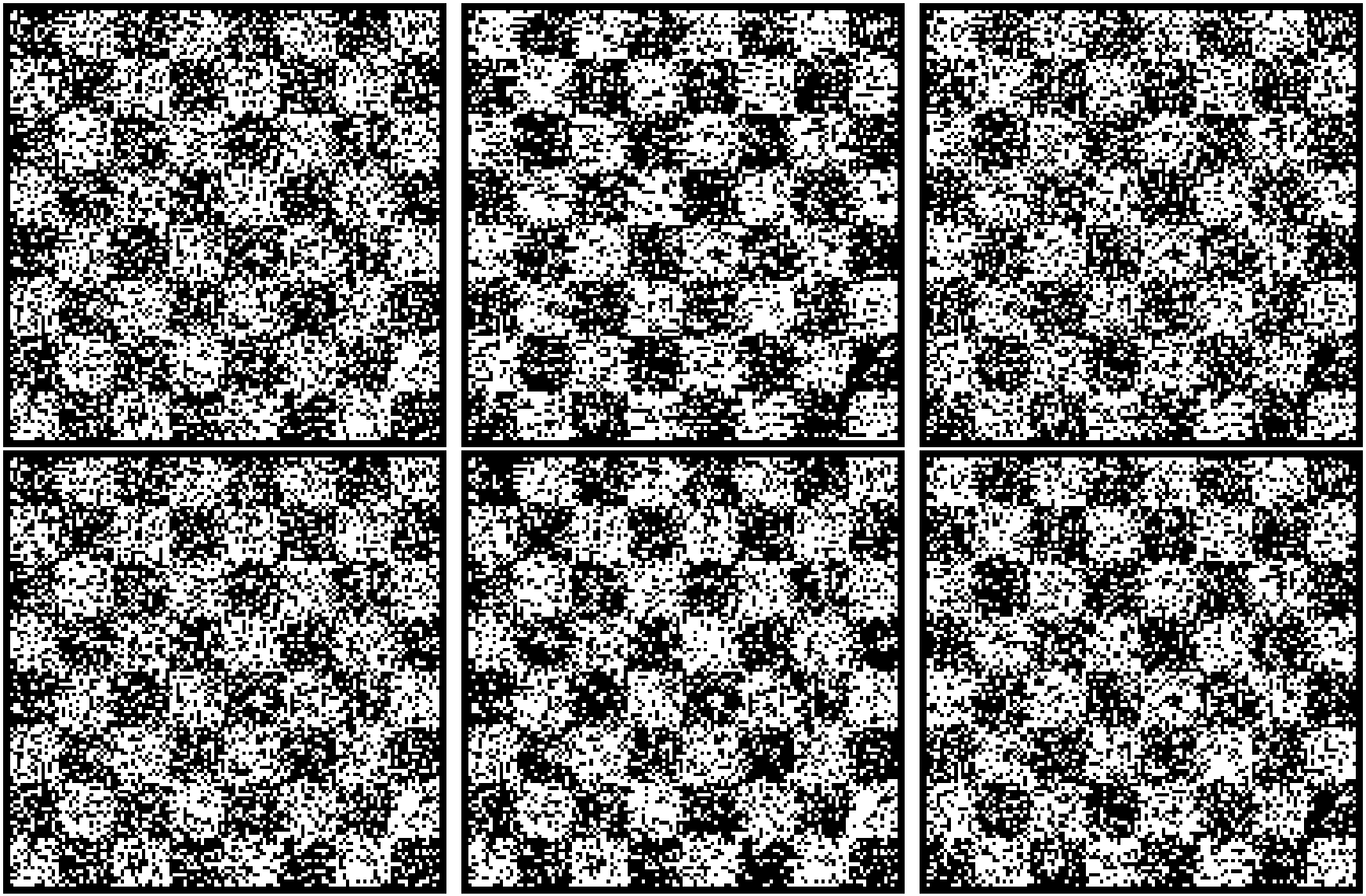
Considering a noisy image of a chessboard with only one feature.

1. *Which technique has which property?*

Application to a chessboard



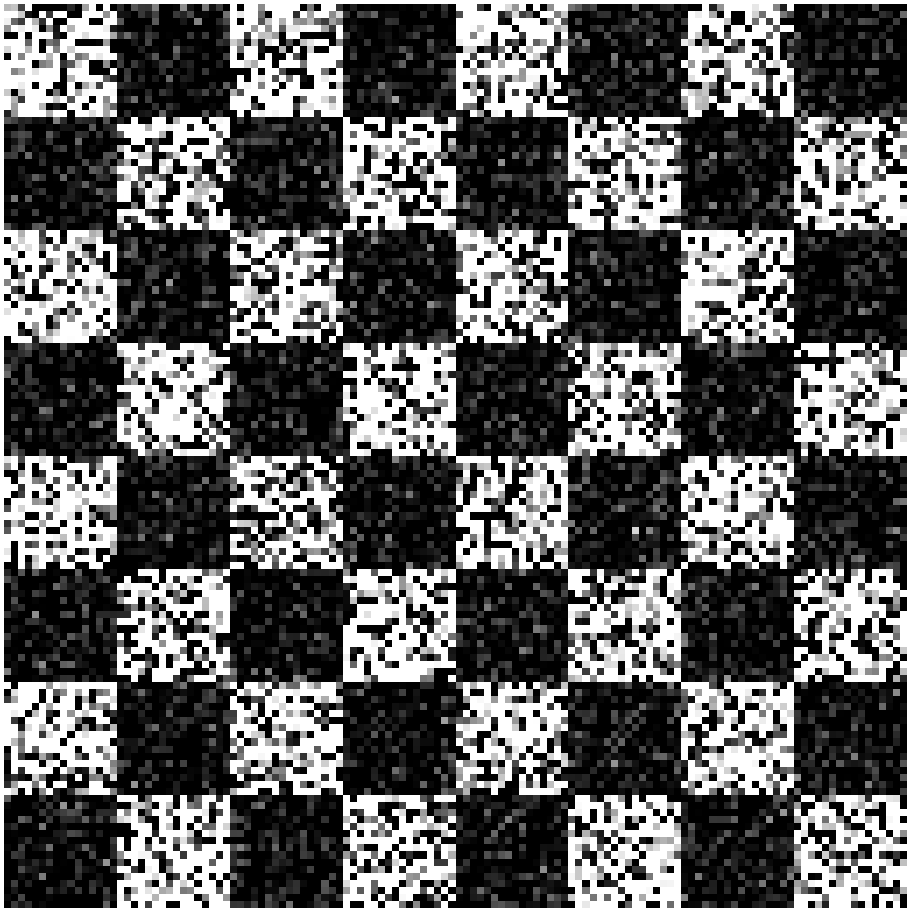
Feature used in the kmeans algorithm in the next slide

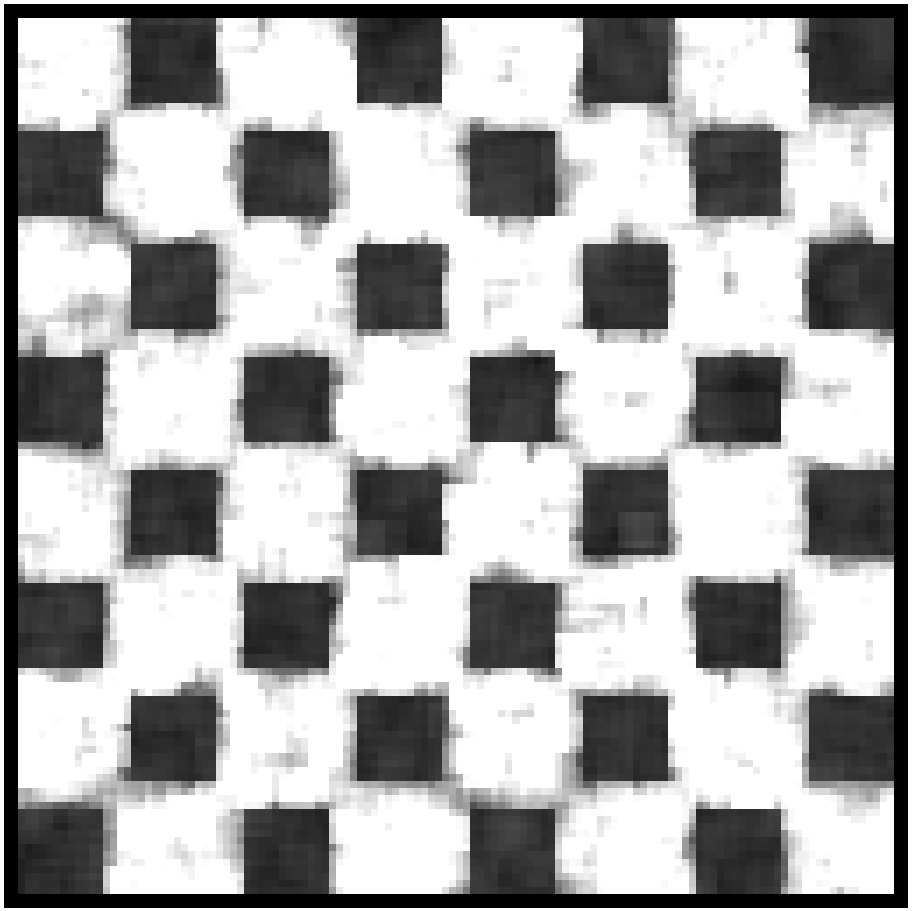


- Pixel value
- Diversity
- Horizontal filtering
- Correlation
- Variance
- Mean

Content of section 5, Spatial context

An example





Explaining the experiment

$$y_{mn} = \text{Chess Board}$$

$$\overset{r}{x}_{mn} \sim \mathcal{N}(y_{mn}, 0.2 + y_{mn})$$

$$\text{noise}_{mn} = \sqrt{\sum_{\mathscr{V}_{mn}} (f_{m'n'} - \mu_{mn})^2}$$

$$\mu_{mn} = \frac{1}{25} \sum_{\mathscr{V}_{mn}} f_{m'n'}$$

An application of noise estimation

The noise measurement is here a measurement specific to the feature. Let us denote these measurements as $\mathbf{z}$ and $\mathbf{Z}$ for the corresponding dataset. $\mathbf{z}$ and $\mathbf{Z}$ are of the same size than $\mathbf{x}$ and $\mathbf{X}$.

A noise-aware PCA algorithm

$$\widehat{\mathbf{e}} = \underset{\mathbf{e},\ \mathbf{e}\mathbf{Z}^T\mathbf{Z}\mathbf{e}^T=1}{\arg\max} \quad \mathbf{e}\mathbf{X}^T\mathbf{X}\mathbf{e}^T$$

This is actually a linear algebra problem called **generalized eigenvalue problem**. An algorithm is to find

$$\widehat{\mathbf{e}} = \underset{\mathbf{e},}{\arg\max}\ \mathbf{e}\mathbf{X}^T\mathbf{X}\mathbf{e}^T - \lambda\mathbf{e}\mathbf{Z}^T\mathbf{Z}\mathbf{e}^T$$

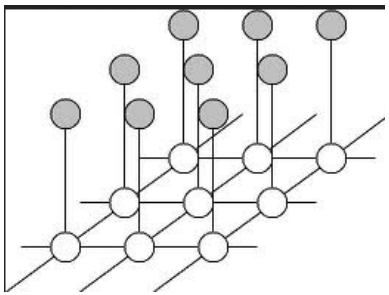
with λ chosen to fit the constraint.

Content of section 5, Spatial context

Conditional random Markov field

Assumption

It is likely that the neighboring pixels belong to the same class.



- Neighborhood = four closest pixels (generally). Here it is denoted $\mathcal{V}''_{mn}$.
- Conditional probability with respect to neighbors is a Gaussian of the difference.
- Markov property = independence with respect to non-neighbors

An example

Problem at stake

$$y_{mn} = \text{chess Board and } \overset{r}{x}_{mn} \sim \mathcal{N}(y_{mn}, 2)$$

Equations

$$P(Y|X) \propto \prod_{mn} f_1(x_{mn}|y_{mn})\, f_2(y_{mn}|y_{\mathcal{V}''(mn)})$$

where

$$f_1(x_{mn}|y_{mn}) \propto e^{-\frac{1}{2\sigma_1^2}(x_{mn}-\mu_0)^2\delta(y_n=0)-\frac{1}{2\sigma_1^2}(x_{mn}-\mu_1)^2\delta(y_n=1)}$$

$$f_2(y_{mn}|y_{\mathcal{V}''_{mn}}) \propto e^{-\frac{1}{2\sigma_2^2}\sum_{m'n'\in\mathcal{V}''}(y_{mn}-y_{m'n'})^2}$$

Finally we get a new global function to minimize

$$J = \sum_{mn}(x_{mn}-\mu_0)^2\delta(y_n=0) + (x_{mn}-\mu_1)^2\delta(y_n=1) + \lambda \sum_{m'n'\in\mathcal{V}''_{mn}} (y_{mn}-y_{m'n'})^2$$

And this time the simulated annealing is clearly not powerful enough.