

Supervised Classification of Hyperspectral Images (HIP2)

March 20, 2024

Gabriel Dauphin

<https://www-l2ti.univ-paris13.fr/~dauphin/HIP2.htm>

Gabriel.dauphin@univ-paris13.fr

(please mention HIP2 in the subject of emails).

Outline I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

8. Supplementary material regarding matrices

Intent of the lessons

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

Out of the scope

- State-of-the-art •
- Neural Networks •
- Ensemble classifiers •

Questions

Feel free to ask questions: chat, end of subsection, at any time...

Figures

- DOI indicates a publication from which the photo is extracted.
- Obtained with Octave, see `lecture_notes.pdf` on `HIP2.htm`

<https://octave.org/>

The packages being used are

- `optim`
- `signal`
- `image`
- `statistics`

Kind of data

- What is it used for?
- Number of dimensions?

Technique

- What is the objective?
- What is the input?
- What is the output?
- Why is it expected to work?

Second level of understanding

Formulas

- What is it computing?
- What parameters it depends on?
- Letters on notations can be misleading: \hat{y}_n is actually not depending on y_n .
- Notations are different depending on the context.

Graphics

- Axis?

Pseudocode

- input/output?
- Number of loops?
- Way out?

Choose tools to make them yours.

- In a real project, techniques have to be adapted and they are not receive a new name.
- Try modify or create examples to see how it works.
- Test computations with numerical simulations.

Table of Contents I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

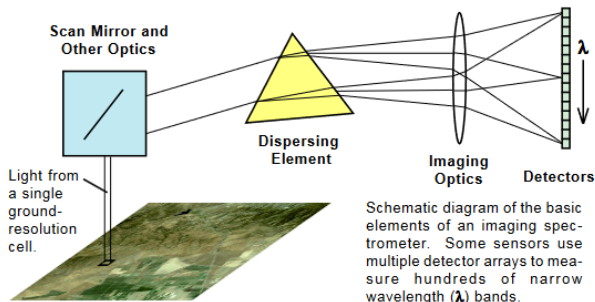
Table of Contents II

8. Supplementary material regarding matrices

Content of section 1, Classification of hyperspectral images

- 1.1 Hyperspectral images
- 1.2 Supervised and unsupervised classification of hyperspectral images
- 1.3 Simple predictors
- 1.4 Accuracy and loss functions
- 1.5 Training, testing and validation sets
- 1.6 Confusion matrix

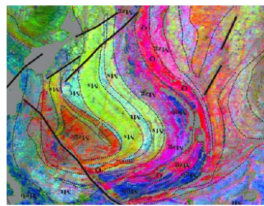
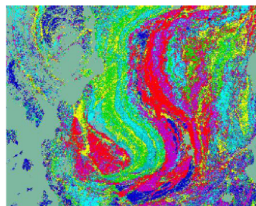
RGB-multispectral-hyperspectral



microimages.com

- RGB
- Collection of wavelengths
- A precise padding of wavelengths

Hyperspectral imaging from space (airborne, drones)



AUG Signals' classified image of 7 materials

Ground truth image provided by the Nature Resource of Canada

■ Blue - veg/metagabbro
■ Dark cyan - metatonalites
■ Light cyan - metatonalites

■ Red - quartzites ■ Green - psammites
■ Magenta monzo - granites
■ Yellow - psammites

CNES 2008

- Finding water?
- Assess forest damages from fire.
- What is being cultivated?
- Level of sea water?
- Collect information through clouds.

Hyperspectral imaging from devices

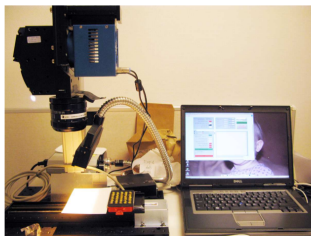


Figure 2.9: Setup of the hyperspectral line-scan NIR camera system with light source and sample conveyor belt.

- Food analysis (quality, non visible rot)
- Recycling: finding constituents

IMM-PHD-2011

What is a color image



$$\left(I_{m_1 m_2}^R, I_{m_1 m_2}^G, I_{m_1 m_2}^B \right)$$

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

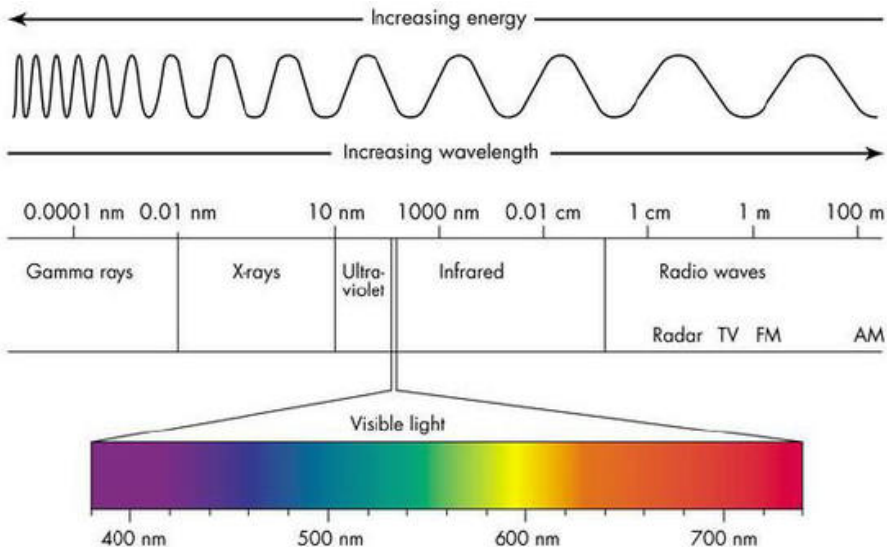
ans(:,:,3) =

    0
    1
    0
```



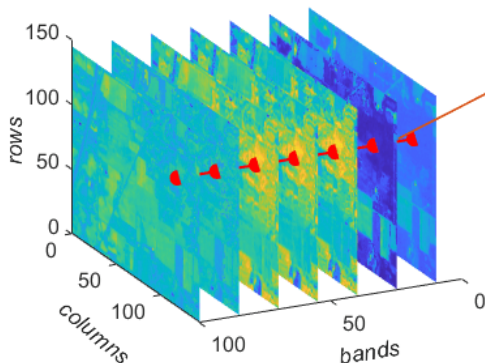
```
R=[1;1;0]; G=[0.5;1;1]; B=[0;1;0];
im=cat(3,R,G,B),
figure(1); imshow(im);
```


Wavelengths

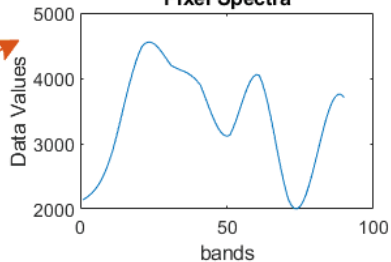


Hyperspectral image

Hyperspectral Data Cube



Pixel Spectra

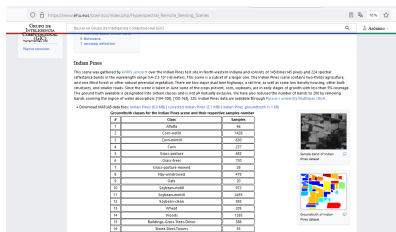


IT implementation

Beware at the frame orientations!

Content of a hyperspectral dataset

- Raw values (set of intensities)
- Corrected values (after registration)
- Ground Truth
- Calibration information



Download links for the Indian Pines scene:

- Download Sentinel-2A Raw Indian Pines (0.1 MB) | Download Indian Pines (0.7 MB) | Indian Pines ground truth (1.1 KB)

Class	Sample
1	100
2	14223
3	1221
4	211
5	452
6	120
7	42
8	419
9	21
10	212
11	1013
12	422
13	229
14	1193
15	329
16	92

Exercise 2

- 1 *Find on the web the hyperspectral image Pine and retrieve it in Octave, using for instance*

```
https://www.ehu.eus/ccwintco/index.php/  
Hyperspectral_Remote_Sensing_Scenes  
https://engineering.purdue.edu/~biehl/  
MultiSpec/hyperspectral.html
```

- 2 *Find the size of each bandwidth image*
- 3 *Find the number of bandwidths*

Answer to exercise 2

① I followed the following steps

- Retrieve `Indian_pines_corrected.mat` and `Indian_pines_gt.mat`
- `T=load` to retrieve the image
- `fieldnames(T)` to get the name of the variable
- `size` to get the answers.

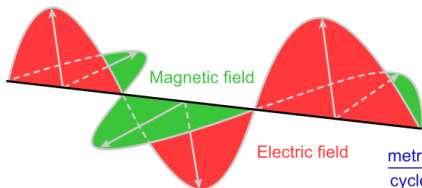
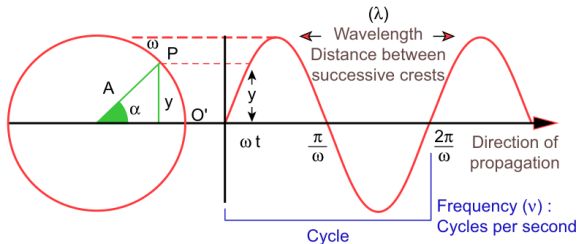
```
ans =  
{  
  [1,1] = indian_pines_corrected  
}
```

```
ans =  
  
    145    145    200
```

② 145×145

③ 200

Wavelengths and frequencies



E. L.

$$\lambda \cdot \nu = c$$

$$\frac{\text{metres}}{\text{cycles}} \cdot \frac{\text{cycles}}{\text{seconds}} = \frac{\text{metres}}{\text{seconds}}$$

E. L.

Doi: 10.1016/B978-0-12-809254-5.00001-4

Exercise 3

- 1 *Retrieve the calibration information*

[https://www.ehu.eus/ccwintco/index.php/
Hyperspectral_Remote_Sensing_Scenes](https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes)

[https://engineering.purdue.edu/~biehl/
MultiSpec/hyperspectral.html](https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html)

- 2 *Considering a horizontal line located at the center of the image, find the coordinate of its left most point.*
- 3 *Plot the spectral intensities as a function of the bandwidths number.*
- 4 *Plot the spectral intensities in terms of radiance and as a function of the center wavelength.*

Keywords

Calibration, registration.

Answer 1/6 to exercise 3

- 1 Using the second website, I went to the following websites

`https://purr.purdue.edu/publications/
1947/1`

`https://purr.purdue.edu/publications/
1947/supportingdocs?v=1`

and got the following text file named

`Calibration_Information_for_220_Channel_Data_Band_Set.txt`

Information on 220 Channel

AVIRIS Data Set

Location

This data is from the AVIRIS

(Airborne Visible/Infrared Imaging Spectrometer)

...

Answer 2/6 to exercise 3

...

These data are calibrated data. In other words the data values in the scene are proportional to radiance. 1000 has been added to the calibrated data so that all data values in this scene are positive. To convert the scene data values (SDV) to radiance values (RV), one must first subtract 1000 and then divide by the gain_factor that JPL used which is 500.

$$RV = (SDV - 1000) / 500.$$

The RV units are $W * cm^{-2} * nm^{-1} * sr^{-1}$.

...

Answer 3/6 to exercise 3

...

AVIRIS Band #	Data Channel #	Center Wavelength (nm)	FWHM (nm)	Center Uncertainty (nm)	FWHM Uncertainty (nm)
1	(not used - the band was all 0's)				
2	1	400.02	9.78	0.92	0.50

Answer 4/6 to exercise 3

- 2 The center horizontal line is at the line number 73:
 $(72 - 1) + 1 = (145 - 74 + 1) = 72$ and $2 \times 72 + 1 = 145$
The left most point is at $m = 73$, $n = 1$.

- 3 It is shown on the left of figure 1.

```
figure(1); plot(im(73,1,:), 'linewidth', 2);
```

- 4 It is shown on the right of figure 1. The following bands have been removed [104 – 108], [150 – 163], 220 as indicated in the dataset because of the water absorption. We get the horizontal scale using the following steps.
- Open the text file
 - Convert each line into arrays of numbers (a line starting with a letter is converted into a void array).
 - Check if the array is non-empty and if its second number is not included in the list of removed bandwidths.
 - Stack in a vector the third component of each non-void array.

Answer 5/6 to exercise 3

We get the vertical scale by making the following affine transform

$$RV = \frac{SDV - 1000}{500}$$

Bandwidths removed

These are indicated with black crosses.

Answer 6/6 to exercise 3

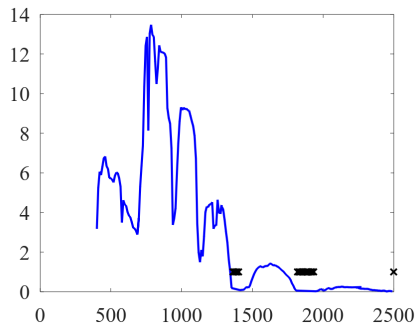
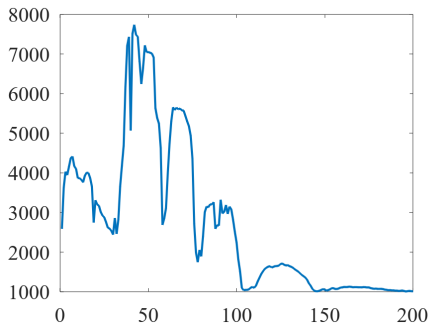
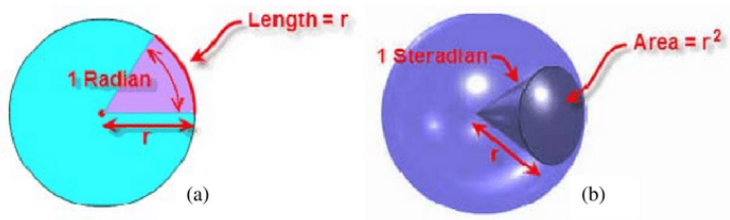


Figure 1: Left: spectral intensities as a function of the bandwidths number. Right: spectral intensities in terms of radiance and as a function of the center wavelength. Exercise 3

What means $W \cdot \text{cm}^{-2} \cdot \text{nm}^{-1} \cdot \text{sr}^{-1}$

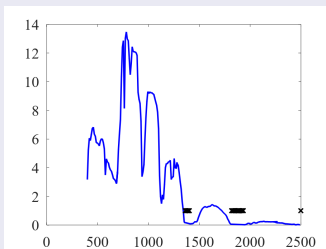
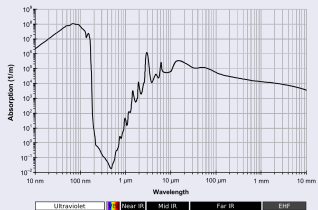
- W : Watt (radiation) is $J/s, N.m.s^{-1}$ or $kg.m^2.s^{-3}$.
- cm^{-2} : surface of $1\text{cm} \times 1\text{cm}$
- nm^{-1} : size of bandwith in $1\text{nm} = 10^{-9}\text{m}$.
- sr^{-1} : measure of angle in 3D: $\in [0, 4\pi]$.



(a) Radian defined (b) Steradian defined

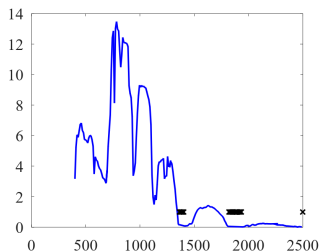
DOI: 10.1117/12.883572

Exercise 4



- 1 Explain the reason for removing the bandwidths indicated with black crosses on the right.

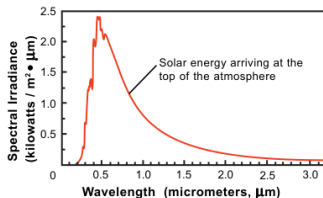
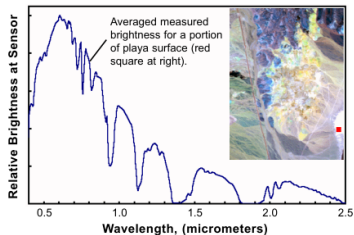
Answer to exercise 4



- lowest wavelength: 400nm
- first set of wavelengths:
 $1362 - 1402\text{nm}$
- second set of wavelengths:
 $1819 - 1893\text{nm}$
- third set of wavelengths:
 2500nm



Water absorbing frequency



microimages.com

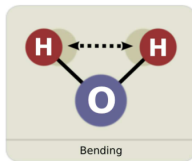
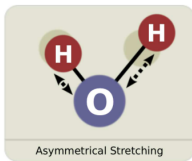
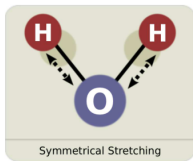


Figure 2.2: The three different vibrational modes of a H_2O molecule. The gray H atom in the background represent the stationary state.

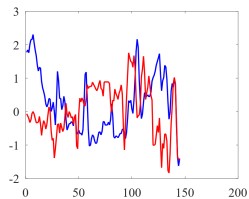
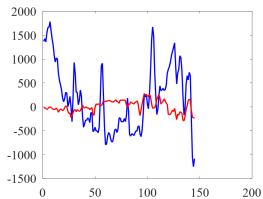
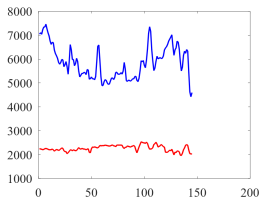
IMM-PHD-2011

Exercise 5

We consider again the central horizontal line. For each questions, use appropriate notations to express the computed quantities.

- 1 Plot the profile line considering the spectral intensity of the bandwidth number 50 and the bandwidth number 100.*
- 2 Center both lines. We here consider that centering assume that pixels at a given bandwidth should be processed in a similar manner.*
- 3 Normalize them so that their variance is equal to one.*

Answer 1/3 to exercise 5



Answer 2/3 to exercise 5

- 1 Let us call

$$m_{1c} = 73 \quad k_1 = 50 \quad k_2 = 100$$

The two profile lines are

$$m_2 \mapsto I(m_{1c}, m_2, k_1) \quad \text{and} \quad m_2 \mapsto I(m_{1c}, m_2, k_2)$$

- 2 The two average intensities are

$$\mu_1 = \frac{1}{M_1 M_2} \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} I(m_1, m_2, k_1)$$

$$\mu_2 = \frac{1}{M_1 M_2} \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} I(m_1, m_2, k_2)$$

The transformed profile lines are

$$m_2 \mapsto I(m_{1c}, m_2, k_1) - \mu_1 \quad \text{and} \quad m_2 \mapsto I(m_{1c}, m_2, k_2) - \mu_2$$

Proof.

$$\sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} (I(m_1, m_2, k_1) - \mu_1) = \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} I(m_1, m_2, k_1) - M_1 M_2 \mu_1 = 0$$

□

Answer 3/3 to exercise 5

- 3 The two standard deviations are

$$\sigma_1 = \sqrt{\frac{1}{M_1 M_2 - 1} \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} (I(m_1, m_2, k_1) - \mu_1)^2}$$

$$\sigma_2 = \sqrt{\frac{1}{M_1 M_2 - 1} \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} (I(m_1, m_2, k_2) - \mu_2)^2}$$

The transformed profile lines are

$$m_2 \mapsto \frac{I(m_{1c}, m_2, k_1) - \mu_1}{\sigma_1} \quad \text{and} \quad m_2 \mapsto \frac{I(m_{1c}, m_2, k_1) - \mu_2}{\sigma_2}$$

Proof.

$$\begin{aligned} & \sqrt{\frac{1}{M_1 M_2} \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} \left(\frac{I(m_1, m_2, k_1) - \mu_1}{\sigma_1} \right)^2} \\ &= \frac{1}{\sigma_1} \sqrt{\frac{1}{M_1 M_2} \sum_{m_1=0}^{M_1-1} \sum_{m_2=0}^{M_2-1} (I(m_1, m_2, k_1) - \mu_1)^2} = 1 \end{aligned}$$



- \mathcal{I} is a hyperspectral image, it is a rank 3 tensor, $I(m_1, m_2, k)$ is a component.
- m_1, m_2, k are the row, column and bandwidth indexes.
- M_1, M_2, K are the number of columns, rows and bandwidths.
- μ is the mean of a set of numbers.
- σ is the standard deviation of a set of numbers.

Conclusion of section 1, Classification of hyperspectral images

- rank 3 tensors
- radiance, wavelengths, spectrum
- registration, calibration
- spatial displacement and wavelength shift

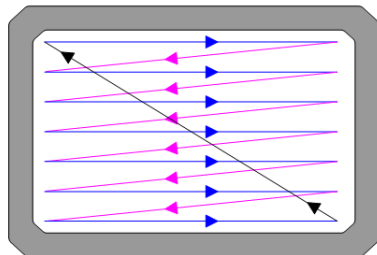
What are classifiers

In the next section, we discuss how to make these images informative.

Content of section 1, Classification of hyperspectral images

- 1.1 Hyperspectral images
- 1.2 Supervised and unsupervised classification of hyperspectral images**
- 1.3 Simple predictors
- 1.4 Accuracy and loss functions
- 1.5 Training, testing and validation sets
- 1.6 Confusion matrix

Raster scanning order

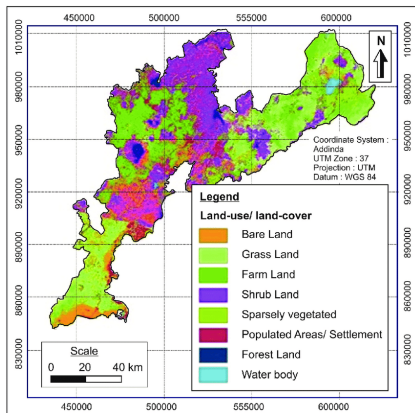


- Feature space $\mathbf{x} \in \mathbb{R}^F$
- Input matrix
- Sample, instance or record \mathbf{x}_n
- Set of samples

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_{N-1} \end{bmatrix} \quad (1)$$

Beware

Often the raster scanning order reads along columns.



- 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

Keywords

Classification is sometimes referred
 to as labelling

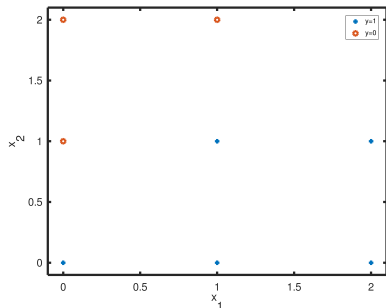
Exercise 6

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

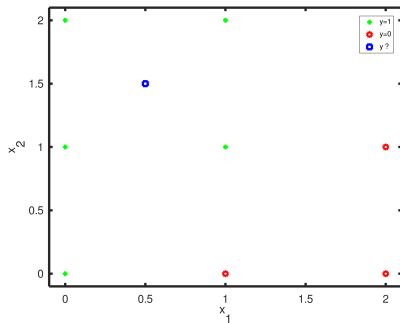
$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 2 & 0 \\ 2 & 1 \\ 2 & 2 \end{bmatrix} \quad Y = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Answer to exercise 6

```
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]);
```

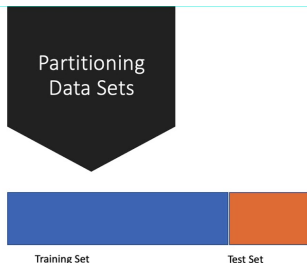


What is classification?



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

Training and testing set



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

- Training set
- Test set
- Supervised classification problem

Parameter estimation

Training set \Rightarrow Parameters } \Rightarrow Accuracy
Testing set \Rightarrow Testing set }

Ground truth: a challenging issue

- It is a hard work to go every where to collect the information.
- Some locations can be difficult to access.
- Definitions of labels may not be appropriate to what is actually going on.
- The time at which the hyperspectral image is recorded may not match that of the ground truth.

True applications

It is acknowledged that many datasets have up 20 % mislabeled samples.

Number of samples per class in training set and testing set

year	min training	max training	min testing	max testing	DOI
2018	10	113	906	11158	10.1109/TCYB .2019 .2905793
2010	15	50	5	2418	10.1109/LGRS .2010 .2047711
2008	$\frac{N}{3} \frac{1}{6}$	$\frac{N}{3} \frac{1}{6}$	$\frac{2N}{3} \frac{1}{6}$	$\frac{2N}{3} \frac{1}{6}$	10.1007/978 -3-540 -85567-5.52
2000	$\frac{N}{5} \frac{1}{16}$	$\frac{N}{5} \frac{1}{16}$	$\frac{4N}{5} \frac{1}{16}$	$\frac{4N}{5} \frac{1}{16}$	10.1109/IGARSS .2000 .861712

Challenging issue

Imbalance dataset (a.k.a variations in class abundance)

Keywords

- Supervised classification
- Semi-supervised classification
- Unsupervised classification a.k.a. clustering

Exercise 7



Figure 2: Classification map indicating in white the soybean.

- 1 Denoting \mathcal{C} the collection of classes that are soybean, $0 \dots C - 1$ the total set of classes, write the pseudo-code of an algorithm yielding figure 6

Answer to exercise 7

- 1
 - \mathcal{C} is the set of requested labels.
 - l_{gd} is groundtruth map.
 - l_c is the yielded classification map.

Require: $\mathcal{C}, l_{\text{gd}}$

Ensure: l_c

- 1: Set l_c to the size of l_{gd} with null values.
- 2: **for** $n \in \mathcal{I}$ **do**
- 3: **if** $l_{\text{gd}}(n) \in \mathcal{C}$ **then**
- 4: $l_c(n) = 1$

New notations

- lower case indicates scalars: $f_{m_1 m_2}$, except $I_{m_1 m_2}$.
- Bold lower case indicates row vectors: \mathbf{x} .
- Capital letters indicate column vectors: Y .
- Bold capital letters indicate matrices: \mathbf{X} , \mathbf{I} .
- Sets are in calligraphic fonts: \mathcal{C} , \mathcal{N} .
- $n \in \{0 \dots N - 1\}$ is the index of sample \mathbf{x}_n .
- Image intensities are here considered as a data set x_n
- Bandwidths are now considered as features $\mathbf{x} = [x_0 \dots x_{F-1}]$.
- Land use and land covers are indicated with $y_n \in \{0 \dots C - 1\}$.
- Ground truth map and classification map: \mathbf{I}_{gd} , \mathbf{I}_{c} .

Conclusion of subsection 2, Supervised and unsupervised 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.

Content of section 1, Classification of hyperspectral images

- 1.1 Hyperspectral images
- 1.2 Supervised and unsupervised classification of hyperspectral images
- 1.3 Simple predictors**
- 1.4 Accuracy and loss functions
- 1.5 Training, testing and validation sets
- 1.6 Confusion matrix

Predictor function

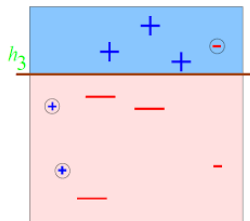
- Predictor function

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

- Iverson bracket

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

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



Exercise 8

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

$f_{a,b}(x) = (2a - 1)1(x \leq b) + 1 - a$
with a and b as parameters.

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

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

Integers representing binaries

Logic

-1, +1

0, 1

$y = \text{POSITIVE}$

$$y = +1$$

$$y = 1$$

$y = \text{NEGATIVE}$

$$y = -1$$

$$y = 0$$

$y_1 = y_2$

$$y_1 y_2$$

$$\begin{aligned} & 1(y_1 = y_2) \\ &= y_1 y_2 + (1 - y_1)(1 - y_2) \\ &= (2y_1 - 1)y_2 + (1 - y_1) \\ &= 0.5\tilde{y}_1\tilde{y}_2 + 0.5 \end{aligned}$$

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

Answer to exercise 8

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

$$f_{x,y}(z) = f_{x,z}(y)1(y = z)$$

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

① $f_{1,2}(0.5) = (2 \times 1 - 1)1(0.5 \leq 2) + 1 - 1 = 1$

$f_{1,0.5}(2) = (2 \times 1 - 1)1(2 \leq 0.5) + 1 - 1 = 0$

② Assuming $y = z$, $f_{x,y}(z) = f_{x,z}(z) = f_{x,z}(y)$

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

$$= (2x - 1)(1 - 1(y < z)) + 1 - x = (1 - 2x)1(y \leq z) + x$$

$$= -(2x - 1)1(y \leq z) + 1 - (1 - x) = 1 - f_{x,z}(y)$$

Decision stumps: definition

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

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

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

$$f_{\theta_F, \theta_x, 0}(\mathbf{x}) = 1 - 1(x_{\theta_F} \leq \theta_x) = 1(x_{\theta_F} > \theta_x)$$

$$f_{\theta_F, \theta_x, 1}(\mathbf{x}) = 1(x_{\theta_F} \leq \theta_x)$$

Scalar product

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

$$\mathbf{x} \cdot \mathbf{x}' = \sum_{f=1}^F x_f x'_f \quad \text{and} \quad \|\mathbf{x}\|^2 = \mathbf{x} \cdot \mathbf{x}$$

This scalar product can be written with matrix operations.

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

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

Euclidean distance

$$d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| = \sqrt{(\mathbf{x} - \mathbf{x}')(\mathbf{x} - \mathbf{x}')^T} = \sqrt{\sum_{f=1}^F (x_f - x'_f)^2}$$

Exercise 9

We consider a predictor f defined as

$$f(\mathbf{x}) = 1(2x_1 + x_2 \leq 2) \quad (3)$$

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

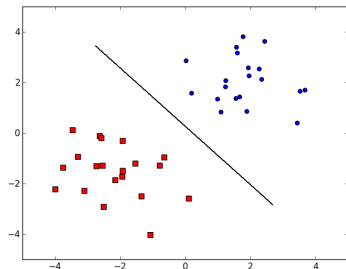
We are considering two sets

$$\mathcal{X}_0 = \{\mathbf{x} \mid f(\mathbf{x}) = 0\} \text{ and } \mathcal{X}_1 = \{\mathbf{x} \mid f(\mathbf{x}) = 1\}$$

- 6 Plot a line separating the two sets and indicate which set is where?

Linear predictors

$$f_{\mathbf{a},b}(\mathbf{x}) = 1(\mathbf{a}\cdot\mathbf{x} \leq b)$$



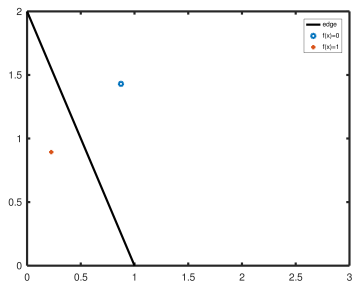
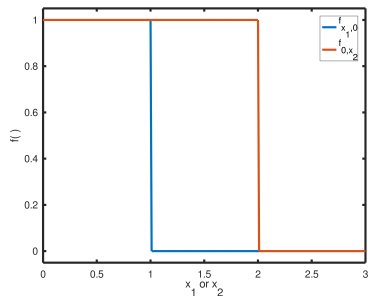
Remark

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 \mathbf{a} to define this predictor, then we need some extra information.

Answer to exercise 9

$$f(\mathbf{x}) = 1(2x_1 + x_2 \leq 2)$$

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



- Predicted output: \hat{y} (it depends on \mathbf{x}).
- 1: Inversion bracket ($1(0 = 1) = 0$ and $1(2 + 2 = 4) = 1$).
- Θ : the whole set of parameters.
- parameters: $\theta_F, \theta_x, \theta_y$.
- Threshold on intensity θ_x .
- \cdot scalar product.
- $\| \cdot \|$ norm of the scalar product.
- \mathbf{x}^T is a column vector and T is the transpose.

Conclusion of subsection 3, Simple predictors

- 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 \hat{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 1, Classification of hyperspectral images

- 1.1 Hyperspectral images
- 1.2 Supervised and unsupervised classification of hyperspectral images
- 1.3 Simple predictors
- 1.4 Accuracy and loss functions**
- 1.5 Training, testing and validation sets
- 1.6 Confusion matrix

- OA: Overall Accuracy

$$OA = \frac{1}{N} \sum_{n=0}^{N-1} 1(y_n = \hat{y}_n)$$

- AA: Average Accuracy

$$AA = \frac{1}{C} \sum_{c=0}^{C-1} \frac{\sum_{n=0}^{N-1} 1(y_n = c) 1(y_n = \hat{y}_n)}{\sum_{n=0}^{N-1} 1(y_n = c)}$$

Notations

Here, accuracies are denoted as A.

Accuracy vs loss functions

Accuracy (overall accuracy)

what is at stake?

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

Example of loss function

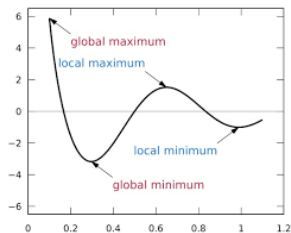
$$L(Y, \hat{Y}) = -A(Y, \hat{Y})$$

In terms of notations, Y and \hat{Y} are column vectors stacking y_n and \hat{y}_n . y_n is the true label and \hat{y}_n is the label predicted using \mathbf{x}_n .

This is actually a simplification.

Note that in $L(Y, \hat{Y})$, \hat{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} .

max, min, argmax, argmin



- Value of the global maximum: $\max_x f(x)$
- Value of the global minimum: $\min_x f(x)$
- Input points of the global maximum: $\operatorname{argmax}_x f(x)$
- Input points of the global minimum: $\operatorname{argmin}_x f(x)$

Exercise 10

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

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

with a and b as parameters. and the following database \mathcal{S}_1

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

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

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

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

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

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

Answer to exercise 10

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

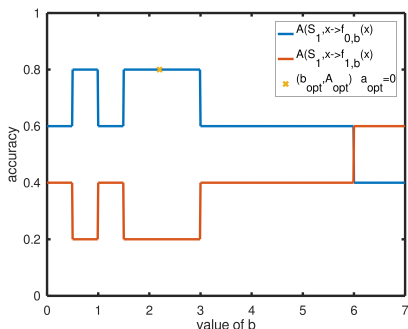
$$x_1 = 1 \quad y_1 = 1$$

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

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

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

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



1 $a = 1$

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

$$1(y_2 = \hat{y}_{2,b}) = 1(b < 1.5)$$

$$1(y_3 = \hat{y}_{3,b}) = 1(b \geq 6)$$

$$1(y_4 = \hat{y}_{4,b}) = 1(b \geq 3)$$

$$1(y_5 = \hat{y}_{5,b}) = 1(b < 0.5)$$

2 $a = 0$

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

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

4 $A_{\text{opt}} = 0.8$.

5

$$(a_{\text{opt}}, b_{\text{opt}}) \in \operatorname{argmax}_{a,b} A(\mathcal{S}_1, f_{a,b})$$

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

Loss-functions used for learning

- Supervised classification: L depends on Y and \hat{Y} or...
- Unsupervised classification: L depends on \mathbf{X} and \hat{Y} .

Learning

Parameters are selected so as to minimize the loss function.

New notations

- Accuracies: OA, AA and A.
- Output and inputs of global extrema: \max , \min , argmin , argmax .
- Loss function: L .
- Labels: Y and \hat{Y} stacking y_n and \hat{y}_n .

Conclusion of subsection 4, Accuracy and loss functions

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

- 1.1 Hyperspectral images
- 1.2 Supervised and unsupervised classification of hyperspectral images
- 1.3 Simple predictors
- 1.4 Accuracy and loss functions
- 1.5 Training, testing and validation sets**
- 1.6 Confusion matrix

Training and testing set

Partitioning
Data Sets



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

- Training set
- Test set
- Supervised classification problem

- $[\mathcal{S}_{\text{TRAIN}}, \mathcal{S}_{\text{TEST}}] = \text{SPLIT}(\mathcal{S}, [\frac{3}{4}, \frac{1}{4}])$
- $\Theta = \text{LEARN}(\mathcal{S})$
- $A = \text{TEST}(\mathcal{S}, \Theta)$

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 $K = 5$.



$$A_{k,p} = \text{TEST}(\text{LEARN}(\mathcal{I}_{k' \neq k}, \theta_p), \mathcal{I}_k)$$
$$\theta_{p_{\text{opt}}} = \underset{p \leq P}{\text{argmin}} \sum_k A_{k,p} \Rightarrow \theta_{\text{opt}}$$

What for?

Cross validation can be used to make decisions based on the dataset, this amounts to using the **validation** set in the red boxes. It can be used to make a more accurate performance measurement, than the use of the **test** set in the red boxes.

Exercise 11

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

① Improve the following algorithm using validation sets.

Require: $\mathcal{S}_3, \mathcal{S}_4$: data sets

Ensure: \mathbf{a}, b : linear classifier

- 1: $\mathcal{S}_{opt} = \mathcal{S}_3$.
- 2: $(\mathbf{a}_{opt}, b_{opt}) = \text{LEARN}(\mathcal{S}_{opt})$
- 3: Compute A_{opt} with $(\mathbf{a}_{opt}, b_{opt})$ and \mathcal{S}_{opt} .
- 4: **repeat**
- 5: $(\mathbf{x}, (\mathbf{x}', y')) = \text{argmin}_{\mathbf{x} \in \mathcal{S}_4, (\mathbf{x}', y') \in \mathcal{S}_3} d(\mathbf{x}', \mathbf{x})$
- 6: Set $\mathcal{S} = \mathcal{S}_{opt} \cup (\mathbf{x}, y')$
- 7: $(\mathbf{a}, b) = \text{LEARN}(\mathcal{S})$
- 8: Compute $A = \text{TEST}(\mathcal{S}, (\mathbf{a}, b))$
- 9: **if** $A > A_{opt}$ **then**
- 10: $(\mathbf{a}_{opt}, b_{opt}) = (\mathbf{a}, b), \mathcal{S}_{opt} = \mathcal{S}, A_{opt} = A$.
- 11: **until** $A \leq A_{opt}$

Answer to exercise 11

Require: $\mathcal{S}_3, \mathcal{S}_4, l$

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

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

Continuation of answer to exercise 11

Require: $\mathcal{S}_3, \mathcal{S}_4$

Ensure: (\mathbf{a}, b)

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

New notations

- Machine learning tools: SPLIT, LEARN, TEST
- optimal value of a parameter: opt.

Conclusion of subsection 5, 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 know 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 5, 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 1, Classification of hyperspectral images

- 1.1 Hyperspectral images
- 1.2 Supervised and unsupervised classification of hyperspectral images
- 1.3 Simple predictors
- 1.4 Accuracy and loss functions
- 1.5 Training, testing and validation sets
- 1.6 Confusion matrix**

Confusion matrix

Predicted Labels



True Labels



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

Confusion matrix

Components of the confusion matrix

$$\mathbf{C} = [c_{ij}] \text{ and } c_{ij} = \sum_{n=0}^{N-1} 1(y_n = i)1(\hat{y}_n = j)$$

- **Here** i, j are index of classes.
- y_n true class of sample number n .
- \hat{y}_n predicted class of sample n .
- N total number of samples (here not the number of rows).

Beware

Sometimes, rows and columns are swapped in this definition.

Exercise 12

We consider the following confusion matrix.

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

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

Answer to exercise 12

1

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

2 6×6 .

$$[1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1]$$

\vdots

$$[1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0]$$

$$[0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0]$$

\vdots

\vdots

$$[0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0]$$

Exercise 13

We are considering the following matrix

$$\mathbf{C} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 2 & 4 \end{bmatrix}$$

- 1 How many classes are there?
- 2 How many samples have been tested?
- 3 Up to some renumbering, what are the values of y_n ?
- 4 Using the same ordering, what are the values of \hat{y}_n ?
- 5 Compute the OA?
- 6 Compute the AA?
- 7 Show that $OA = \frac{c_{00} + c_{11} + c_{22}}{N}$.
- 8 Show that $AA = \frac{1}{3} \left(\frac{c_{00}}{c_{00} + c_{01} + c_{02}} + \frac{c_{11}}{c_{10} + c_{11} + c_{12}} + \frac{c_{22}}{c_{20} + c_{21} + c_{22}} \right)$

Answer to exercise 13 I

- 1 $C = 3$ because the size of \mathbf{C} is 3×3 .
- 2 $N = 13$ because $N = \sum_{ij} C_{ij}$
- 3 Let us reading the lines of \mathbf{C} .
 - 4 samples are part of the class 0: $y_0 = y_1 = y_2 = y_3 = 0$.
 - 3 samples are part of the class 1: $y_4 = y_5 = y_6 = 1$.
 - 6 samples are part of the class 2:
 $y_7 = y_8 = y_9 = y_{10} = y_{11} = y_{12} = y_{13} = 2$.
- 4 Let us read the columns of \mathbf{C} .
 - 2 samples have been predicted as being part of the class 0:
 $\hat{y}_0 = \hat{y}_1 = 0$.
 - 4 samples have been predicted as being part of the class 1:
 $\hat{y}_2 = \hat{y}_4 = \hat{y}_7 = \hat{y}_8 = 1$.
 - 7 samples have been predicted as being part of the class 2:
 $\hat{y}_3 = \hat{y}_5 = \hat{y}_6 = \hat{y}_9 = \hat{y}_{10} = \hat{y}_{11} = \hat{y}_{12} = \hat{y}_{13} = 2$.
- 5 Let us consider the diagonal components of \mathbf{C}

Answer to exercise 13 II

- Among the 4 samples part of class 0, there are 2 correct predictions:
 $y_0 = \hat{y}_0$ and $y_1 = \hat{y}_1$.
- Among the 3 samples part of class 1, there is 1 correct predictions:
 $y_4 = \hat{y}_4$.
- Among the 6 samples part of class 2, there are 4 correct predictions:
 $y_{10} = \hat{y}_{10}$, $y_{11} = \hat{y}_{11}$, $y_{12} = \hat{y}_{12}$ and $y_{13} = \hat{y}_{13}$.

$$\text{OA} = \frac{2 + 1 + 4}{13} = \frac{6}{13}$$

$$\text{AA} = \frac{1}{3} \left(\frac{2}{4} + \frac{1}{3} + \frac{4}{6} \right) = \frac{1}{2}$$

$$\begin{aligned} \text{OA} &= \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{1}(y_n = \hat{y}_n) \\ &= \frac{1}{N} \sum_{c=0}^{C-1} \sum_{n=0}^{N-1} \mathbf{1}(y_n = c) \mathbf{1}(y_n = \hat{y}_n) = \frac{C_{00} + C_{11} + C_{22}}{N} \end{aligned}$$

8

$$\begin{aligned}AA &= \frac{1}{C} \sum_{c=0}^{C-1} \frac{\sum_{n=0}^{N-1} \mathbf{1}(y_n=c) \mathbf{1}(y_n=\hat{y}_n)}{\sum_{n=0}^{N-1} \mathbf{1}(y_n=c)} \\ &= \frac{1}{C} \sum_{c=0}^{C-1} \frac{C_c}{C_{c0} + C_{c1} + C_{c2}} \\ &= \frac{1}{3} \left(\frac{C_{00}}{C_{00} + C_{01} + C_{02}} + \frac{C_{11}}{C_{10} + C_{11} + C_{12}} + \frac{C_{22}}{C_{20} + C_{21} + C_{22}} \right)\end{aligned}$$

- Confusion matrix $\mathbf{C} = [c_{ij}]$.
- Column vector of predicted labels: \hat{Y} .

Conclusion of subsection 6, 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.

Table of Contents I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

Table of Contents II

8. Supplementary material regarding matrices

Content of section 2, Image processing I

2.1 Segmentation

2.2 Edges as a mean for segmentation

2.3 Detection of connected components

2.4 Use of iterated algorithms

2.5 Clustering regarded as an optimization problem

Segmentation is a partition of the pixels in subsets.

$$\mathcal{N} = \bigcup_c \mathcal{C}_c \text{ s.t. } \mathcal{C}_c \cap \mathcal{C}_{c'} = \emptyset \quad (4)$$

Each set contains pixels that are homogeneous in some sense.

- Point-based techniques using only one bandwidth.

Thresholding

A set of pixels can be defined by thresholding w.r. to T .

$$\mathcal{C} = \{(m_1, m_2) \in \mathcal{N} \mid I(m_1, m_2) \geq T\}$$

Superlevel set

$T \mapsto \mathcal{C}$ is called the superlevel set. It is also related to the empirical distribution $F(T) = 1 - \frac{1}{N}|\mathcal{C}|$.

We are going to consider the cardinality of this set.

How can we select T ?

Exercise 14

To investigate the choice of the threshold, we are investigating the properties of the following curves. Given an image \mathbf{I} , let $f_{\mathbf{I}}$ be defined as

$$f_{\mathbf{I}}(T) = |\{n \in \mathcal{N} \mid I(n) \geq T\}|$$

- 1 *Is $f_{\mathbf{I}}$ increasing, decreasing, or...?*

Exercise

- 2 Compute $f_I(0)$, $\lim_{+\infty} f_I$

Let I_r be the centered and normalized image I and f_{I_r} the corresponding function.

$$I_r(n) = I(n) - \mu \quad \text{where} \quad \mu = \frac{1}{N} \sum_{n=0}^{N-1} I(n)$$

- 3 What is the relation between f_I and f_{I_r} ?

Answer to exercise 14 I

- ① $f_{\mathbf{I}}$ is decreasing: let $T_1 < T_2$.

$$\{n | I(n) \geq T_1\} \subset \{n | I(n) \geq T_2\} \Rightarrow f_{\mathbf{I}}(T_1) \geq f_{\mathbf{I}}(T_2)$$

②

$$\{n | I(n) \geq 0\} = \mathcal{N} \Rightarrow f_{\mathbf{I}}(0) = |\mathcal{N}| = N$$

$$\forall T > \max_n I(n), \quad \{n | I(n) \geq T\} = \emptyset \Rightarrow \lim_{+\infty} f_{\mathbf{I}} = |\emptyset| = 0$$

③

$$f_{\mathbf{I}_r}(T) = |\{n | I_r(n) \geq T\}| = |\{n | I(n) - \mu \geq T\}| = f_{\mathbf{I}}(T + \mu)$$

The curve is moved too the **left**.

For unsupervised binary classification, a possible loss-function is

$$L(\mathbf{X}, \hat{Y}) = \sum_{\hat{y}_n=0, \hat{y}_{n'}=0} \|\mathbf{x}_n - \mathbf{x}_{n'}\|^2 + \sum_{\hat{y}_n=1, \hat{y}_{n'}=1} \|\mathbf{x}_n - \mathbf{x}_{n'}\|^2$$

Exercise 15

Based on the definition of a decision stump in machine learning and using the L2-loss function applied to real valued predictors, how could a threshold be computed?

Answer to exercise 15 I

- We consider two families of decision stumps: $f_a(\mathbf{X}) = 1(x_{nf} \leq T)$ and $f_b(\mathbf{X}) = 1(x_{nf} > T)$.

- Accordingly, we define two loss-functions L and L'

$$L(f, T) = \sum_{n=0}^{N-1} \sum_{n'=0}^{N-1} \|\mathbf{x}_n - \mathbf{x}_{n'}\|^2 f_a(\mathbf{x}_n) f_a(\mathbf{x}_{n'}) \\ + \sum_{n=0}^{N-1} \sum_{n'=0}^{N-1} \|\mathbf{x}_n - \mathbf{x}_{n'}\|^2 (1 - f_a(\mathbf{x}_n))(1 - f_a(\mathbf{x}_{n'}))$$

L' is defined using f_b instead of f_a .

- For each f , we compute T_f and T'_f

$$T_f \in \underset{T}{\operatorname{argmin}} L(f, T) \text{ and } T'_f \in \underset{T}{\operatorname{argmin}} L'(f, T)$$

- Finally if $\min_f L(f, T_f) \leq \min_f L'(f, T'_f)$, the proposed decision stump is

$$1(x_{n\hat{f}} \leq T_{\hat{f}}) \text{ with } \hat{f} \in \underset{f}{\operatorname{argmin}} L(f, T_f)$$

If not, then it is

$$1(x_{n\hat{f}} > T'_{\hat{f}}) \text{ with } \hat{f} \in \underset{f}{\operatorname{argmin}} L'(f, T'_f)$$

Median and quantiles

Given a set of values,

$$\mathbf{x} = [15, 6, 13, 8, 8, 10, 7, 3, 16, 20]^T$$

we first reorder them

$$\mathbf{x}_{\text{ord}} = [3, 6, 7, 8, 8, 10, 13, 15, 16, 20]^T$$

- Median is the average value between the fifth and the sixth value:

$$\frac{8+10}{2} = 9$$

$$\lceil 10/2 \rceil = \lceil 5 \rceil = 5$$

- First quartile is the third: 7

$$\lceil 10/4 \rceil = \lceil 2.5 \rceil = 3$$

- Third quartile is the eighth: 15

$$\lceil 3 \times 10/4 \rceil = \lceil 7.5 \rceil = 8$$

A rough approximation of the k -th q -quantile is

$$x_{\text{ord}}[\lceil Np \rceil] \text{ where } N = |\mathbf{x}| \text{ and } p = kq$$

Use of f_I to find an adequate threshold

$$\tau \in f_I^{-1}(p) = \left\{ \tau \mid \exists \mathcal{N}_\tau \text{ such that } \begin{array}{l} I(n) \leq \tau \Leftrightarrow n \in \mathcal{N}_\tau \\ |\mathcal{N}_\tau| = p \end{array} \right\}$$

Rounding notations

$$3 = \lfloor 3.4 \rfloor = \lfloor 3.4 \rfloor < \lceil 3.4 \rceil = 4$$

$$3 = \lfloor 3.6 \rfloor < \lfloor 3.6 \rfloor = \lceil 3.6 \rceil = 4$$

$$-4 = \lfloor -3.4 \rfloor < \lfloor -3.4 \rfloor = \lceil -3.4 \rceil = -3$$

$$-4 = \lfloor -3.6 \rfloor = \lfloor -3.6 \rfloor < \lceil 3.6 \rceil = -3$$

Simulation result

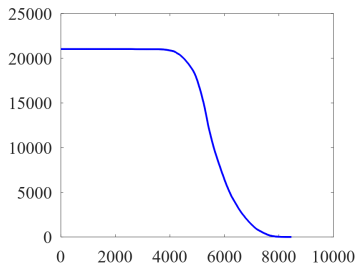


Figure 3: Example of f_1 -function as defined in exercise 14 for the Indian's Pine hyperspectral image using the bandwidth number 50.

Exercise 16

- 1 Looking at figure 3, what does it tell us on the hyperspectral image?
- 2 Show on figure 3, the first, second and third quartiles.

Distance induced partitions

Euclidean distance in the spectral space \mathbb{R}^K

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{f=0}^F (a_f - b_f)^2} = \|\mathbf{a} - \mathbf{b}\|$$

With two points $\mathbf{a}, \mathbf{b} \in \mathbb{R}^F$, we get a segmentation of \mathbf{I}

$$\mathcal{N}_{\mathbf{a}} = \{n \in \mathcal{N} \mid d(I_n, \mathbf{a}) < d(I_n, \mathbf{b})\}$$

$$\mathcal{N}_{\mathbf{b}} = \{n \in \mathcal{N} \mid d(I_n, \mathbf{a}) \geq d(I_n, \mathbf{b})\}$$

Exercise 17

We consider two sets $\mathcal{N}_{\mathbf{a}}$ and $\mathcal{N}_{\mathbf{b}}$ defined as the set of pixels being closer to \mathbf{a}, \mathbf{b} than of \mathbf{b}, \mathbf{a} .

- 1 Show that $\mathcal{N}_{\mathbf{a}}$ and $\mathcal{N}_{\mathbf{b}}$ are segmentations of \mathbf{I} in the sense of equation (4).

We denote \mathbf{X} the dataset obtained using the intensities of \mathcal{I} at the different bandwidths as defined in equation (1).

- 2 Show that there exists U and b such that $1(\mathbf{X}U \leq b)$ is a binary column vector indicating the membership of each row to $\mathcal{N}_{\mathbf{a}}$. Show that $1(\mathbf{X}U > b)$ indicates that of $\mathcal{N}_{\mathbf{b}}$.

Answer to exercise 17 I

- ① \mathcal{N}_a and \mathcal{N}_b are a partition of \mathcal{N} .
- $\mathcal{N}_a \cap \mathcal{N}_b = \emptyset$ because we cannot have both $d(I_n, \mathbf{a}) < d(I_n, \mathbf{b})$ and $d(I_n, \mathbf{a}) \geq d(I_n, \mathbf{b})$
 - $\mathcal{N}_a \subset \mathcal{N}$ and $\mathcal{N}_b \subset \mathcal{N}$.
 - $\mathcal{N} \subset \mathcal{N}_a \cup \mathcal{N}_b$ because either $d(I_n, \mathbf{a}) < d(I_n, \mathbf{b})$ is true or $d(I_n, \mathbf{a}) \geq d(I_n, \mathbf{b})$ is true.

- ② Considering the scalar product \cdot ,

$$d^2(\mathbf{x}, \mathbf{a}) - d^2(\mathbf{x}, \mathbf{b}) = \|\mathbf{x} - \mathbf{a}\|^2 - \|\mathbf{x} - \mathbf{b}\|^2$$

$$= (\mathbf{x} - \mathbf{a} + \mathbf{x} - \mathbf{b}) \cdot (\mathbf{x} - \mathbf{a} - \mathbf{x} + \mathbf{b}) = 2 \left(\mathbf{x} \cdot (\mathbf{b} - \mathbf{a}) - \frac{\|\mathbf{b}\|^2 - \|\mathbf{a}\|^2}{2} \right)$$

Therefore we set $U = (\mathbf{b} - \mathbf{a})^T$ and $b = \frac{\|\mathbf{b}\|^2 - \|\mathbf{a}\|^2}{2}$. \mathcal{N}_b is the complement of \mathcal{N}_a .

- Image, slices and components: \mathcal{I} , \mathbf{I} instead of \mathbf{I}_k , $I(n)$ instead of $I(n, k)$
- $f_{\mathbf{I}}$ and $f_{\mathbf{I}}^{-1}$, using $\{\dots | \dots\}$ to define a set.
- Sets of pixels: \mathcal{N} , $\mathcal{N}_{\mathbf{a}}$, $\mathcal{N}_{\mathbf{b}}$ and \mathcal{C}_c .
- \cup , \cap , \emptyset , partition, \subset .
- Cardinality of a set: $|\mathcal{S}|$
- Rounding notations: $\lfloor \dots \rfloor$, $\lceil \dots \rceil$, $\lceil \dots \rceil$.

Conclusion of subsection 1, Segmentation

- We have seen common issues between unsupervised classification and segmentation.
- We have defined a loss function for binary unsupervised classification problems.
- We defined a function f useful to select thresholds and it happens to be the superlevel set function.
- Thresholding can be seen as a decision stump.
- With two points, we define a linear classifier.

What are the tools in image processing to consider the spatial context?

Nearby points tend to belong to similar classes.

Actually there are also links with probability through the empirical distribution and the empirical cumulative distribution.

Content of section 2, Image processing I

- 2.1 Segmentation
- 2.2 Edges as a mean for segmentation
- 2.3 Detection of connected components
- 2.4 Use of iterated algorithms
- 2.5 Clustering regarded as an optimization problem

Edge detection

Roberts operators

Convolution is here the sum-product along a sliding window.

$$F_1 = \begin{bmatrix} \boxed{1} & 0 \\ 0 & -1 \end{bmatrix} \quad \text{and} \quad F_2 = \begin{bmatrix} \boxed{0} & 1 \\ -1 & 0 \end{bmatrix}$$

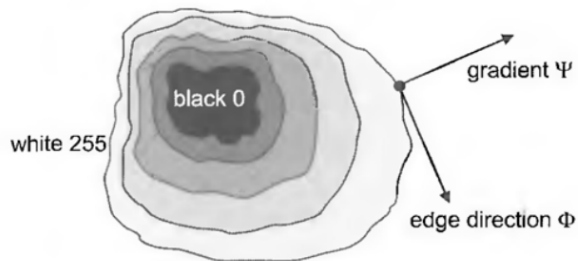
The magnitude of the edge is

$$|F_1 * I| + |F_2 * I|$$

This is equivalent to

$$\begin{aligned} |\partial I(m_1, m_2)| &= |I(m_1, m_2) - I(m_1 + 1, m_2 + 1)| \\ &\quad + |I(m_1, m_2 + 1) - I(m_1 + 1, m_2)| \end{aligned}$$

Detection of the edge angle



DOI: 10.1007/978-1-4899-3216-7

Detection of the edge angle

Two operators are added

$$F_3 = \begin{bmatrix} \boxed{1} & -1 \end{bmatrix} \quad \text{and} \quad F_4 = \begin{bmatrix} \boxed{1} \\ -1 \end{bmatrix}$$

The angle of the steepest increase (or gradient) is $\psi \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$, that of the edge angle is Φ .

$$[\Psi(m_1, m_2), \Phi(m_1, m_2)] = \begin{cases} -\frac{3\pi}{4} & \frac{3\pi}{4} & \text{if } [(F_2 * I)(m_1, m_2)]_- \geq F_{\max} \\ -\frac{\pi}{2} & \pi & \text{if } [(F_4 * I)(m_1, m_2)]_- \geq F_{\max} \\ -\frac{\pi}{4} & -\frac{3\pi}{4} & \text{if } [(F_1 * I)(m_1, m_2)]_- \geq F_{\max} \\ 0 & -\frac{\pi}{2} & \text{if } [(F_3 * I)(m_1, m_2)]_- \geq F_{\max} \\ \frac{\pi}{4} & -\frac{\pi}{4} & \text{if } [(F_2 * I)(m_1, m_2)]_+ \geq F_{\max} \\ \frac{\pi}{2} & 0 & \text{if } [(F_4 * I)(m_1, m_2)]_+ \geq F_{\max} \\ \frac{3\pi}{4} & \frac{\pi}{4} & \text{if } [(F_1 * I)(m_1, m_2)]_+ \geq F_{\max} \\ \pi & \frac{\pi}{2} & \text{if } [(F_3 * I)(m_1, m_2)]_+ \geq F_{\max} \end{cases}$$

where

$$x = x_+ - x_- \text{ where } x_+ = x1(x \geq 0) \text{ and } x_- = |x|1(x \leq 0)$$

and

$$F_{\max} = \max_c |(F_c * I)(m_1, m_2)|$$

Exercise 18

We consider the following image

$$I = \begin{bmatrix} 1 & 6 & 3 & 3 \\ 2 & 6 & 2 & 4 \\ 1 & 1 & 1 & 5 \\ 5 & 6 & 4 & 1 \end{bmatrix}$$

- 1 Compute the resulting edge-image obtained with the magnitude of the gradient obtained using the Roberts operators.
- 2 Compute the angle of the edge detector.

Answer to exercise 18 I

1

$$\begin{bmatrix} \boxed{1} & 0 \\ 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 & 6 & 3 & 3 \\ 2 & 6 & 2 & 4 \\ 1 & 1 & 1 & 5 \\ 5 & 6 & 4 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} (1 \times 1 + 0 \times 2 & 0 \times 6 + -1 \times 6) & (1 \times 6 + 0 \times 2 & 0 \times 3 + -1 \times 4) & (1 \times 3 + 0 \times 3 & 0 \times 3 + -1 \times 4) \dots \\ \dots \\ \dots \\ \dots \end{bmatrix}$$

$$F_1 * I = \begin{bmatrix} -5 & 4 & -1 & 3 \\ 1 & 5 & -3 & 4 \\ -5 & -3 & 0 & 5 \\ 5 & 6 & 4 & 1 \end{bmatrix} \quad \text{and} \quad F_2 * I = \begin{bmatrix} 4 & -3 & 2 & -4 \\ 5 & 1 & 3 & -5 \\ -4 & -5 & 1 & -1 \\ 6 & 4 & 1 & 0 \end{bmatrix}$$

$$F_3 * I = \begin{bmatrix} -5 & 3 & -1 & 4 \\ -4 & 4 & -2 & 4 \\ 0 & 0 & -4 & 5 \\ -1 & 2 & -3 & 1 \end{bmatrix} \quad \text{and} \quad F_4 * I = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 1 & 5 & 1 & -1 \\ -4 & -5 & -3 & 4 \\ 5 & 6 & 1 & 0 \end{bmatrix}$$

Answer to exercise 18 II

The resulting magnitude of the gradient is

$$\begin{bmatrix} 15 & 10 & 5 & 11 \\ 11 & 15 & 9 & 14 \\ 13 & 13 & 8 & 15 \\ 17 & 18 & 12 & 3 \end{bmatrix}$$

2 The angles obtained are

$$\begin{bmatrix} \{-\frac{\pi}{4}, 0\} & \frac{3\pi}{4} & \pi & \{0, \frac{3\pi}{4}, \pi\} \\ \pi & \{\frac{3\pi}{4}, \frac{\pi}{2}\} & \{-\frac{\pi}{4}, \pi\} & 0 \\ -\frac{\pi}{4} & -\frac{\pi}{2} & 0 & \{\frac{3\pi}{4}, \pi\} \\ \pi & \{\frac{3\pi}{4}, \frac{\pi}{2}\} & \{\frac{3\pi}{4}, \frac{\pi}{2}\} & \{\frac{\pi}{2}, \frac{3\pi}{4}, \pi\} \end{bmatrix}$$

Binomial filter approximating a 2D-Gaussian.

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

Example of two edge maps I

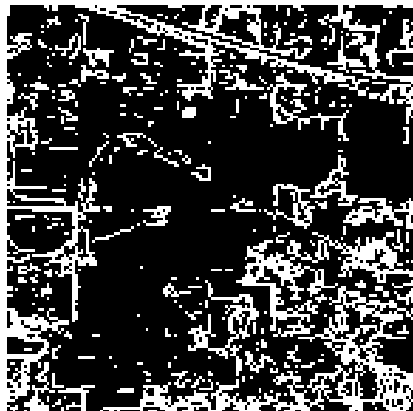


fig13.m, fig12.m

Example of two edge maps II

Require: I

Ensure: I'

- 1: Compute M
- 2: Compute $\tau = f_M^{-1}(\lfloor 0.75N \rfloor)$
- 3: Compute $I' = 1(M \geq \tau)$

$$M = |F_1 * I| + |F_2 * I| + |F_3 * I| + |F_4 * I|$$

$$M' = |F_1 * G * I| + |F_2 * G * I| + |F_3 * G * I| + |F_4 * G * I|$$

Require: I

Ensure: I'

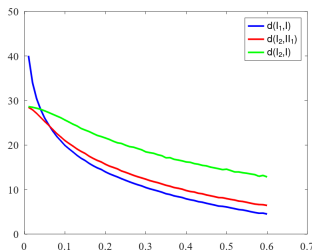
- 1: Compute M'
- 2: Compute $\tau = f_{M'}^{-1}(\lfloor 0.75N \rfloor)$
- 3: Compute $I' = 1(M' \geq \tau)$

Main idea

- Smooth the image
- Compute Gradient
- Add absolute values
- Compute a threshold

Smoothing=denoising

- Require \mathbf{I}
- Normalize between 0 and 1
- Add white Gaussian noise
 $\Rightarrow \mathbf{I}_1$
- Smooth $\Rightarrow \mathbf{I}_2$
- Compare with \mathbf{I}



This

$$\text{PSNR}_{\text{dB}}(\mathbf{I}, \mathbf{I}') = 10 \log_{10} \left(\frac{N}{\sum_{n=0}^{N-1} (I(n) - I'(n))^2} \right)$$

proves \mathbf{I}_2 is closer to \mathbf{I} than to \mathbf{I}_1 , thanks to smoothing.

\log_{10}

$$\log_{10}(10) = 1 \quad \log_{10}(10^x) = x \quad \log_{10}(x) = \frac{\ln(x)}{\ln(10)}$$

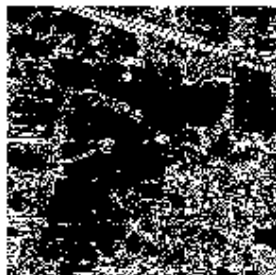
What smoothing removes contains information



Ground truth



$$T = f_I^{-1}(0.75N)$$
$$1(I \geq T)$$



$$I' = |I - \frac{1}{16}G * I|$$
$$T = f_I^{-1}(0.75N)$$
$$1(I \geq T)$$

Texture

To classify texture, we can use nonlinear filters resembling to denoising and we can use entropy-based metrics.

New notations

- ∂I is here a contour, that is a binary image.
- Ψ and Φ are here images whose values are angles.
- $*$ is here the 2D-convolution product. It is actually a sum-product of a sliding window. It uses here \square to indicate how the sliding window is to be positioned.
- $|\dots|$ means the absolute value when applied to a numerical value or function.
- Four examples of filtering operators to find edges: F_1, F_2, F_3, F_4 .
- One example of a smoothing operator reducing noise: G .
- PSNR_{dB} is a metric used in image processing.
- \log_{10} .

Conclusion of subsection 2, Edges as a mean for segmentation

- Filtering operators can smooth the image and reduce noise.
- We have defined a loss function for binary unsupervised classification problems.
- We defined a function f_t useful to select thresholds and it happens to be the superlevel set function.
- Thresholding can be seen as a decision stump.
- With two points, we define a linear classifier.

What are the tools in image processing to consider the spatial context?

Nearby points tend to belong to similar classes.

Actually there are also links with probability through the empirical distribution and the empirical cumulative distribution.

Content of section 2, Image processing I

- 2.1 Segmentation
- 2.2 Edges as a mean for segmentation
- 2.3 Detection of connected components**
- 2.4 Use of iterated algorithms
- 2.5 Clustering regarded as an optimization problem

Connected spaces in mathematics

- $(1, 3)$ is connected but $\{x \mid x \in (1, 3) \text{ or } x = 4\}$ is not connected.
- \mathbb{R} is connected but \mathbb{N} is not connected.
- \mathbb{R}^2 is connected but not

$$\{(x, y) \in \mathbb{R}^2 \mid x + y \neq 0\}$$

This is based on neighborhoods which contain balls.

$$\forall \mathbf{x}, \exists \epsilon > 0 \text{ such that } \mathcal{N}_{\mathbf{x}} \supset \{\mathbf{x}' \mid d(\mathbf{x}, \mathbf{x}') \leq \epsilon\}$$

Connected sets in image processing

A set of pixels is binary image

$$I_{\mathcal{P}}(m_1, m_2) = 1 \text{ } ((m_1, m_2) \in \mathcal{P})$$

These two first are connected sets, not the last one

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

We define a neighborhood system.

$$\mathcal{N}_{(m_1, m_2)} = \{(m'_1, m'_2) \in \mathcal{N} \mid |m'_1 - m_1| + |m'_2 - m_2| \leq 1\}$$

We may also consider as a neighborhood system.

$$\mathcal{N}'_{(m_1, m_2)} = \{(m'_1, m'_2) \in \mathcal{N} \mid |m'_1 - m_1| \leq 1 \text{ and } |m'_2 - m_2| \leq 1\}$$

Definition of connected spaces

\mathcal{P} is connected if

For all $(m_1, m_2), (m'_1, m'_2) \in \mathcal{P}$, there exists a sequence $\gamma_1 \dots \gamma_P$ such that

$$\begin{cases} \gamma_1 = (m_1, m_2) \\ \gamma_P = (m'_1, m'_2) \\ \forall p \in \{1 \dots P\}, \gamma_p \in \mathcal{P} \\ \forall p \in \{1 \dots P-1\}, \gamma_{p+1} \in \mathcal{N}_{\gamma_p} \end{cases}$$

\mathcal{P} has P connected components if

There exists $\mathcal{P}_1, \dots, \mathcal{P}_P$ sets that define a partition of \mathcal{P} such that

$$\begin{cases} \mathcal{P} = \bigcup_p \mathcal{P}_p \\ \forall p \neq p', \mathcal{P}_p \cap \mathcal{P}_{p'} = \emptyset \\ \forall p, \mathcal{P}_p \text{ is connected} \\ \forall p \neq p', \mathcal{P}_p \cup \mathcal{P}_{p'} \text{ is not connected} \end{cases}$$

Finding the connected components

Require: I

Ensure: I'

- 1: Set $I' = 0$ with the size of I
- 2: Set $\text{highest} = 0$
- 3: **for** $(m_1, m_2) \in \mathcal{N}$ with raster scanning along lines **do**
- 4: **if** $I'(m_1, m_2) = 0$ **then**
- 5: continue the for-loop
- 6: Collect the two labels above and left
- 7: **if** no labels are collected **then**
- 8: $\text{highest}+ = 1, I'(m_1, m_2) = \text{highest}$ and continue
- 9: **if** one label is collected or two equal labels **then**
- 10: Set $I'(m_1, m_2)$ with the label nearby and continue
- 11: Set $I'(m_1, m_2)$ with lowest collected label
- 12: Change the highest collected label in I' to the lowest.

Exercise 19

- 1 *What neighborhood system is the pseudocode using?*
- 2 *How can we use the pseudocode to test if a given set is connected?*
- 3 *Give the intermediate values of V' when V is defined as*

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

Answer to exercise 19 I

- 1 The cross neighborhood, that is the first one.
- 2 If the yielded image has at most one label, then the set is connected. If not it is not connected.

3

$$\mathbf{I} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathbf{I}'_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 0 & 3 & 3 & 0 \end{bmatrix}$$

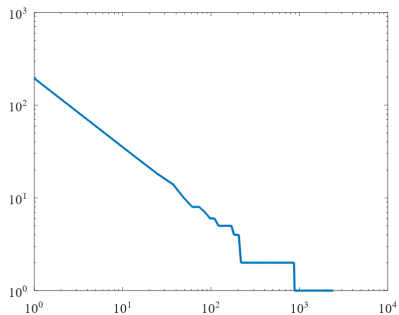
$$\mathbf{I}'_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 \\ 0 & 2 & 2 & 2 \end{bmatrix}$$

Answer to exercise 19 II

Experimental results



Connected components of $1(I(n, 50) \geq \tau)$ with the third quantile.



Number of connected components whose area is greater than a given area.

Experimental results



Ground truth for soybean

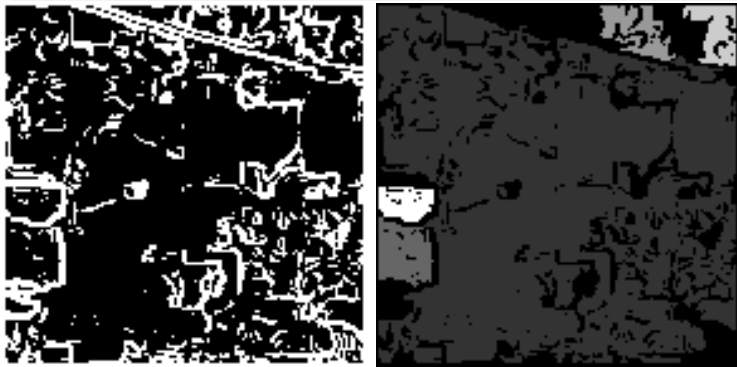


Five greatest connected components.

Edges can be transformed into segmentation

Exercise 20

Give a pseudo transforming a binary image with edges into the corresponding regions.



Answer to exercise 20 I

Require: I_B

Ensure: I'

- 1: Compute $I = 1 - I_B$
- 2: Find the connected components of I .

New notations

- \mathbb{R} , \mathbb{N}
- \forall and \exists

Conclusion of subsection 3, Detection of connected components

- Neighborhood,
- Connected components
- Algorithm giving the connected components
- Edges can be used for region segmentation

Experimental results

The results are not convincing. There is a need for reliable information.

Content of section 2, Image processing I

- 2.1 Segmentation
- 2.2 Edges as a mean for segmentation
- 2.3 Detection of connected components
- 2.4 Use of iterated algorithms**
- 2.5 Clustering regarded as an optimization problem

Setting a thresholding value

seg_thresholding1.m

Require: I

Ensure: T

- 1: select an initial value for T,
- 2: **while** T is modified **do**
- 3:
$$\mu_0 = \frac{\sum_{m_1, m_2} I(m_1, m_2) \mathbf{1}(I(m_1, m_2) \leq T)}{\sum_{m_1, m_2} \mathbf{1}(I(m_1, m_2) \leq T)}$$
- 4:
$$\mu_1 = \frac{\sum_{m_1, m_2} I(m_1, m_2) \mathbf{1}(I(m_1, m_2) \geq T)}{\sum_{m_1, m_2} \mathbf{1}(I(m_1, m_2) \geq T)}$$
- 5:
$$T = \frac{\mu_0 + \mu_1}{2}$$

An initial value of T could be the average between the corners and the center. The algorithm would remain the same if the pixel intensities were stacked in a column vector.

Note

This is a *crisp* assignment.

Exercise 21

We consider the following image

$$\mathcal{I} = \begin{bmatrix} 1 & 6 & 3 & 3 \\ 2 & 6 & 2 & 4 \\ 1 & 1 & 1 & 5 \\ 5 & 6 & 4 & 1 \end{bmatrix}$$

- 1 Give the segmented image using the thresholding algorithm.

Answer to exercise 21 I

- $T = \frac{1+3+5+1}{4} = 2.5$
- $\mu_0 = \frac{1+2+2+1+1+1+1}{7} \approx 1.3$
- $\mu_1 = \frac{6+3+3+6+4+5+5+6+4}{9} \approx 4.6$
- $T = \frac{\mu_1 + \mu_2}{2} \approx 2.97 < 3$

Minimizing the within-diversity

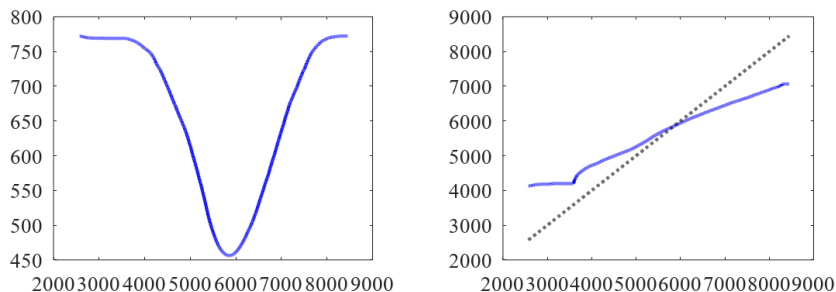


Figure 4: Left: loss function w.r. to T . Right: new threshold T' w.r. to old threshold T .

$$J(\mu_0, \mu_1, \mathcal{N}_0, \mathcal{N}_1) = \sqrt{\frac{\sum_{n \in \mathcal{N}_0} (I_n - \mu_0)^2 + \sum_{n \in \mathcal{N}_1} (I_n - \mu_1)^2}{|\mathcal{N}|}}$$

Minimizing the within-diversity

Exercise 22

We are going to prove formulas in steps 3 and 4 used in algorithm `seg_thresholding1`. We assume a function to be minimized

$$J = \sqrt{\frac{\sum_{n \in \mathcal{N}_0} (I_n - \mu_0)^2 + \sum_{n \in \mathcal{N}_1} (I_n - \mu_1)^2}{N}}$$

- ① Show that given \mathcal{N}_0 and \mathcal{N}_1 , J is minimal when

$$\mu_0 = \frac{\sum_{n \in \mathcal{N}_0} I_n}{|\mathcal{N}_0|} \text{ and } \mu_1 = \frac{\sum_{n \in \mathcal{N}_1} I_n}{|\mathcal{N}_1|}$$

- ② Show that given μ_0 and μ_1 , J is minimal when

$$\mathcal{N}_0 = \left\{ I_n \leq \frac{\mu_0 + \mu_1}{2} \right\} \text{ and } \mathcal{N}_1 = \left\{ I_n > \frac{\mu_0 + \mu_1}{2} \right\}$$

Good news

We don't need to estimate here σ .

Answer to exercise 22 I

- ① Because \mathcal{N}_0 and \mathcal{N}_1 are fixed, the minimization of J is the same than that of $J^2(|\mathcal{N}_0|^2 + |\mathcal{N}_1|^2)$

$$J^2(|\mathcal{N}_0|^2 + |\mathcal{N}_1|^2) = |\mathcal{N}_0| \sum_{n \in \mathcal{N}_0} (I_n - \mu_0)^2 + |\mathcal{N}_1| \sum_{n \in \mathcal{N}_1} (I_n - \mu_1)^2$$

Both quantities are second order polynomials with μ_0 and μ_1 as variables. Considering the left part of the $J^2(|\mathcal{N}_0|^2 + |\mathcal{N}_1|^2)$:

$$\begin{aligned} |\mathcal{N}_0| \sum_{n \in \mathcal{N}_0} (I_n - \mu_0)^2 &= |\mathcal{N}_0| \sum I_n^2 - 2\mu_0 |\mathcal{N}_0| \sum I_n + \mu_0^2 |\mathcal{N}_0|^2 \\ &= |\mathcal{N}_0|^2 \left(\mu_0 - \frac{1}{|\mathcal{N}_0|} \sum I_n \right)^2 + |\mathcal{N}_0| \sum I_n^2 - |\mathcal{N}_0|^2 \left(\frac{1}{|\mathcal{N}_0|} \sum_n x_n \right)^2 \end{aligned}$$

Looking at this equation, the right part is not depending on μ_0 , and the left part is minimized when $\mu_0 = \frac{1}{|\mathcal{N}_0|} \sum_{n \in \mathcal{N}_0} I_n$. Applied on the right part of $J^2(|\mathcal{N}_0|^2 + |\mathcal{N}_1|^2)$, the same technique shows

$$\mu_1 = \frac{1}{|\mathcal{N}_1|} \sum_{n \in \mathcal{N}_1} I_n.$$

General technique

The minimization of J w.r. to μ_0, μ_1 is usually solved using

$$\frac{\partial J}{\partial \mu_0} = 0 \text{ and } \frac{\partial J}{\partial \mu_1} = 0$$

or actually here $\frac{\partial J}{\partial \mu_0}$

- 2 J can be written in a sample-by-sample formula.

$$J^2 = \frac{1}{N} \sum_{n=0}^{N-1} (I_n - \mu_0)^2 \mathbf{1}(n \in \mathcal{N}_0) + (I_n - \mu_1)^2 \mathbf{1}(n \in \mathcal{N}_1)$$

We assume here that $\mu_0 < \mu_1$. Given μ_0, μ_1 and N , J^2 is minimal when for all $n \in \mathcal{N}$

$$\begin{cases} n \in \mathcal{N}_0 & \text{if } |I_n - \mu_0| < |I_n - \mu_1| \\ n \in \mathcal{N}_1 & \text{if } |I_n - \mu_0| \geq |I_n - \mu_1| \end{cases}$$

which is equivalent to

$$\begin{cases} n \in \mathcal{N}_0 & \text{if } x_n \leq \mu_0 \\ n \in \mathcal{N}_0 & \text{if } \mu_0 < x_n \leq (\mu_0 + \mu_1)/2 \\ n \in \mathcal{N}_1 & \text{if } (\mu_0 + \mu_1)/2 < x_n \leq \mu_1 \\ n \in \mathcal{N}_1 & \text{if } x_n > \mu_1 \end{cases}$$

Finally we get

$$\mathcal{N}_0 = \{n | I_n \leq \frac{\mu_0 + \mu_1}{2}\} \text{ and } \mathcal{N}_1 = \{n | I_n > \frac{\mu_0 + \mu_1}{2}\}$$

Using soft assignments

seg_thresholding2.m

Require: $I(n)$

Ensure: μ_0, μ_1

1: select an initial value for T ,

$$2: \mu_0 = \frac{\sum_n I(n) \mathbf{1}(I(n) \leq T)}{\sum_n \mathbf{1}(I(n) \leq T)}$$

$$3: \mu_1 = \frac{\sum_n I(n) \mathbf{1}(I(n) > T)}{\sum_n \mathbf{1}(I(n) > T)}$$

4: **while** μ_0 or μ_1 are modified **do**

$$5: q_n = \frac{|I(n) - \mu_0|}{|I(n) - \mu_0| + |I(n) - \mu_1|}$$

$$6: \mu_0 = \frac{\sum_n I(n) q_n}{\sum_n q_n}$$

$$7: \mu_1 = \frac{\sum_n I(n) (1 - q_n)}{\sum_n (1 - q_n)}$$

Exercise 23

Find a formula for q_n such that the second algorithm behaves like the first one.

Answer to exercise 23 I

$$q_n = 1(I(n) \leq \frac{\mu_0 + \mu_1}{2})$$

$$\frac{\sum_n I(n) q_n}{\sum_n q_n} = \frac{\sum_n I(n) 1(I(n) \leq \frac{\mu_0 + \mu_1}{2})}{\sum_n 1(I(n) \leq \frac{\mu_0 + \mu_1}{2})}$$

$$\frac{\sum_n I(n) (1 - q_n)}{\sum_n (1 - q_n)} = \frac{\sum_n I(n) 1(I(n) > \frac{\mu_0 + \mu_1}{2})}{\sum_n 1(I(n) > \frac{\mu_0 + \mu_1}{2})}$$



Figure 5: Left: Original bandwidth intensity image. Center: Thresholding with crisp assignments. Right: Thresholding with soft assignment.

Is the extra information reliable?

How should $I(n) \mapsto q_n$ be chosen.

$$q_n = \frac{|I(n) - \mu_0|}{|I(n) - \mu_0| + |I(n) - \mu_1|}$$

- $\sum_{n \in \mathcal{N}_0} I(n) = \sum_n I(n) 1(n \in \mathcal{N}_0)$
- The average is $\mu_0 = \frac{1}{|\mathcal{N}_0|} \sum_{n \in \mathcal{N}_0} I(n)$

Conclusion of subsection 4, Use of iterated algorithms

- Thresholding with crisp assignments
- Thresholding with soft assignments
- Graph of a loss function and of a sequence $u_{n+1} = f(u_n)$
- Convexity
- $\frac{\partial}{\partial \theta}$ to the extrema of a function.

What are the tools in image processing to consider the spatial context?

Nearby points tend to belong to similar classes.

Actually there are also links with probability through the empirical distribution and the empirical cumulative distribution.

Content of section 2, Image processing I

- 2.1 Segmentation
- 2.2 Edges as a mean for segmentation
- 2.3 Detection of connected components
- 2.4 Use of iterated algorithms
- 2.5 Clustering regarded as an optimization problem

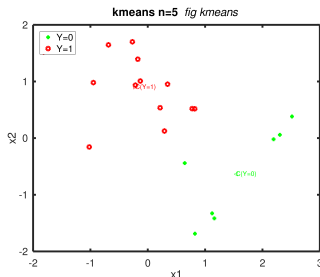
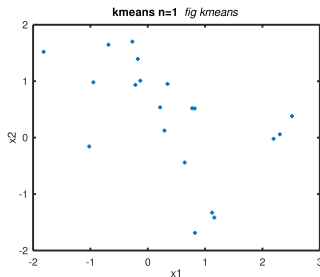
Unsupervised classification

Definition

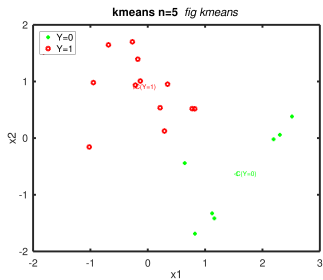
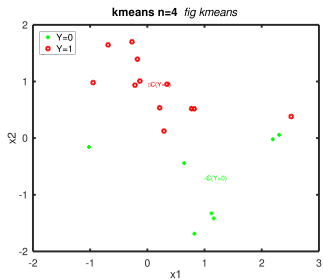
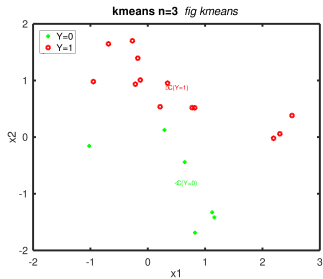
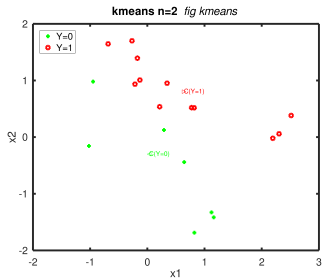
Of the dataset (\mathbf{X}, Y) , only \mathbf{x} is used. ($\mathbf{X}^T = [\mathbf{x}_1^T, \dots, \mathbf{x}_N^T]$)

Clusters

Instead of classes, we consider clusters.



kmeans



Exercise 24

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 24

Require: \mathbf{X}

Ensure: \hat{Y}

- 1: Select randomly two rows of \mathbf{x} : μ_0 and μ_1 .
- 2: Set \hat{Y} with zeros.
- 3: **repeat**
- 4: $\hat{Y}_{\text{old}} = \hat{Y}$
- 5: **for** $n = 1 : N$ **do**
- 6: $\hat{Y}_n = 1(d(\mathbf{x}_n, \mu_0) > d(\mathbf{x}_n, \mu_1))$
- 7: $\hat{Y} = [\hat{Y}_1 \dots \hat{Y}_N]^T$
- 8: $\mu_0 = \frac{1}{|\{n|\hat{Y}_n=0\}|} \sum_{\hat{Y}_n=0} \mathbf{x}_n$
- 9: $\mu_1 = \frac{1}{|\{n|\hat{Y}_n=1\}|} \sum_{\hat{Y}_n=1} \mathbf{x}_n$
- 10: **until** $\hat{Y} = \hat{Y}_{\text{old}}$

An ad hoc loss function

The number of samples assigned to each cluster is

$$|\mathcal{N}_0| = \sum_{n=1}^N 1(y_n = 0) = \sum_{n=0}^{N-1} 1 - y_n \text{ and } |\mathcal{N}_1| = \sum_{n=1}^N 1(y_n = 1) = \sum_{n=0}^{N-1} y_n$$

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

$$\mu_0(\mathbf{X}, Y) = \frac{1}{|\mathcal{N}_0|} \sum_{n=0}^{N-1} (1 - y_n) \mathbf{x}_n$$

$$\mu_1(\mathbf{X}, Y) = \frac{1}{|\mathcal{N}_1|} \sum_{n=1}^{N-1} y_n \mathbf{x}_n$$

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

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

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

$$J(\mathbf{X}, Y) = \sum_{n=0}^{N-1} (1 - y_n) \|\mathbf{x}_n - \mu_0(\mathbf{X}, Y)\|^2 + \sum_{n=0}^{N-1} y_n \|\mathbf{x}_n - \mu_1(\mathbf{X}, Y)\|^2$$

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

Conclusion of subsection 5, Clustering 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 I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

Table of Contents II

8. Supplementary material regarding matrices

Content of section 3, Learning regarded as an optimization problem I

- 3.1 Optimization problem
- 3.2 Simulated annealing
- 3.3 Method of least squares

Optimization problem

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

- Parameters are selected so that

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}} L(Y, [f_{\Theta}^v(\mathbf{x}_n)]_n)$$

where $f_{\Theta}^v(\mathbf{x})$ 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

$$f^v(\mathbf{x}) = b - \mathbf{a} \cdot \mathbf{x}$$

- A new L2-loss function

$$L(\mathcal{S}, f^v) = \frac{1}{2} \sum_{n=0}^{N-1} (f^v(\mathbf{x}_n) - \tilde{y}_n)^2$$

Exercise 25

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

$$x_{11} = 2 \quad x_{12} = 0.5 \quad y_1 = 1$$

$$x_{21} = 1 \quad x_{22} = 2 \quad y_2 = 0$$

$$x_{31} = 0 \quad x_{32} = 0 \quad y_3 = 1$$

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

$$f_{\mathbf{a},b}(\mathbf{x}) = 1(\mathbf{a} \cdot \mathbf{x} \leq b)$$

with $\mathbf{a} = [a_1, a_2]$.

We define $J(a_1, a_2, b) = L(\mathcal{S}_2, f_{\mathbf{a},b})$

① Compute $J(a_1, a_2, b)$ as the sum of three quadratic expressions. And explain why 0 an obvious lower bound of J is likely to be reached.

② Show that $J(a_1, a_2, b) = 0$ 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 $b = 1$.

1

$$J(b, \mathbf{a}) = L(\mathcal{S}, f^v) = \frac{1}{2} \sum_{n=1}^3 (b - \mathbf{a} \cdot \mathbf{x} - \tilde{y}_n)^2$$

Square values are necessarily non-negative so $J(b, \mathbf{a}) \geq 0$. 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

$$2J(b, \mathbf{a}) = (b - 2a_1 - 0.5a_2 - 1)^2 + (b - a_1 - 2a_2 + 1)^2 + (b - 1)^2$$

$J(b, \mathbf{a}) = 0$ iff

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

3

$$\begin{aligned} J(1, [-2/7, 8/7]) &= (1 - 2 * (-2/7) - 0.5 * 8/7 - 1)^2 \\ &\quad + (1 - (-2/7) - 2(8/7) + 1)^2 + (1 - 1)^2 = 0 \end{aligned}$$

Need of a more general technique

In the example shown in exercise 25, we have three samples and three free variables

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

In general this is not true.

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

New notations

- L is here the L2-loss function.
- $f^v(\mathbf{x}) \in \mathbb{R}$ whereas $f(\mathbf{x}) \in \{0, 1\}$.

Conclusion of subsection 1, 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 L_2 -loss function is an example.
- It works like a regression, as if we wanted to predict a real value for \tilde{y} .

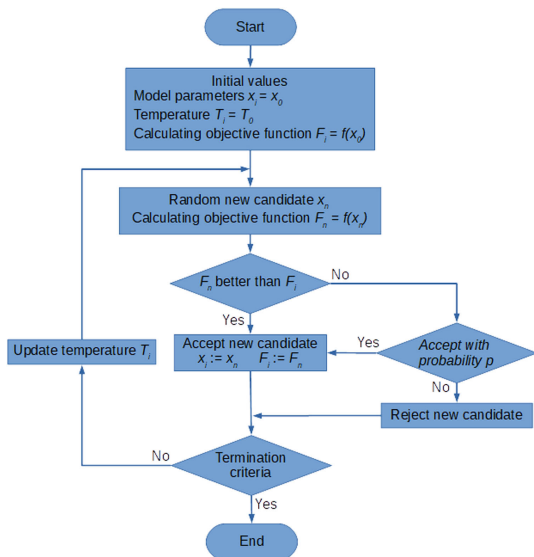
Even a simple example seems to require complex computations, how are we going to deal with more complex examples?

In the next section, we will see an example of algorithm. And after, we will see examples of image processing without optimization.

Content of section 3, Learning regarded as an optimization problem I

- 3.1 Optimization problem
- 3.2 Simulated annealing
- 3.3 Method of least squares

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** $L(\Theta + \Delta\Theta) < L$ **then**
- 6: Set $\Theta := \Theta + \Delta\Theta$ and $L := L(\Theta)$.
- 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 26

Give the Octave code that uses `simulated_annealing` to find an approximation of \mathbf{a} and a of exercise 25 which tells

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

$$x_{11} = 2 \quad x_{12} = 0.5 \quad y_1 = 1$$

$$x_{21} = 1 \quad x_{22} = 2 \quad y_2 = 0$$

$$x_{31} = 0 \quad x_{32} = 0 \quad y_3 = 1$$

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

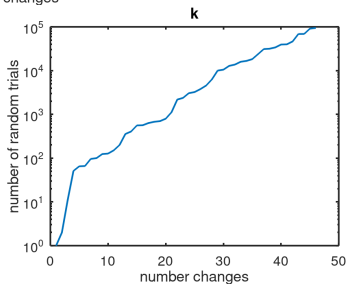
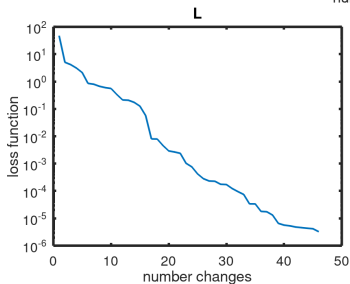
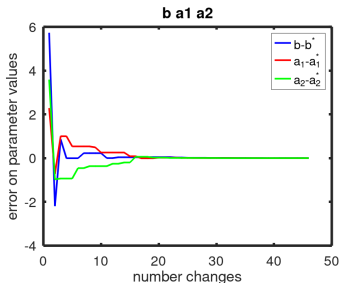
$$f_{\mathbf{a},b}(\mathbf{x}) = 1(\mathbf{a} \cdot \mathbf{x} \leq b)$$

with $\mathbf{a} = [a_1, a_2]$.

We define

$$J(a_1, a_2, b) = L(\mathcal{S}_2, f_{\mathbf{a},b}) = \frac{1}{2} \sum_{n=0}^{N-1} (f^v(\mathbf{x}_n) - \tilde{y}_n)^2$$

Costly performances obtained with simulated annealing



Answer to exercise 26 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);
```

- L is different from L . It is the last best value obtained.
- $\Delta\Theta$: modification of the parameter values.

Conclusion of subsection 2, 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.

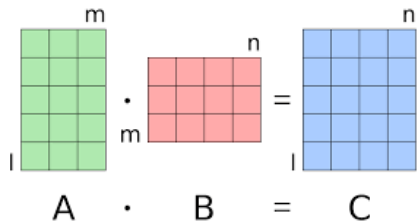
An other technique to select parameters with respect to a loss function and a dataset?

In the next subsection, we discuss the minimization of the L_2 -loss function for linear classifiers.

Content of section 3, Learning regarded as an optimization problem I

- 3.1 Optimization problem
- 3.2 Simulated annealing
- 3.3 Method of least squares

Product of two matrices



$$\begin{bmatrix} 1 & 3 & 2 \\ 3 & 1 & 1 \\ 1 & 2 & 2 \end{bmatrix} \times \begin{bmatrix} 2 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot 2 + 3 \cdot 1 + 2 \cdot 1 & 1 \cdot 1 + 3 \cdot 0 + 2 \cdot 3 & 1 \cdot 1 + 3 \cdot 1 + 2 \cdot 1 \\ 3 \cdot 2 + 1 \cdot 1 + 1 \cdot 1 & 3 \cdot 1 + 3 \cdot 0 + 3 \cdot 2 & 3 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 \\ 2 \cdot 2 + 2 \cdot 1 + 2 \cdot 1 & 1 \cdot 1 + 2 \cdot 0 + 2 \cdot 3 & 1 \cdot 1 + 2 \cdot 1 + 2 \cdot 1 \end{bmatrix}$$

$$C = AB$$

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

Predicting function

$$f^v(\mathbf{x}) = b - \mathbf{a} \cdot \mathbf{x} = \mathbf{w} \cdot \hat{\mathbf{x}}$$

We use the following definition

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

$$\hat{\mathbf{x}} = [x_1 \ x_2 \ \dots \ x_F \ 1] = [\mathbf{x} \ 1]$$

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

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

Expressing the loss function with matrices I

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

Scalar product as vector multiplication

$$\mathbf{w} \cdot \mathbf{x} = \mathbf{w} \mathbf{x}^T$$

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

Sum of square values as vector multiplication

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

In the same way,

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

Expanding follows classical rules

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

Transpose of the product of two matrices

Rules

- $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$ and if \mathbf{AB} is a scalar $\mathbf{AB} = (\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$
- $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$

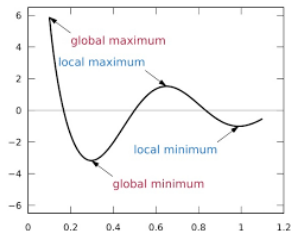
$$2L(\mathcal{S}, f^v) = \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T \right)^T \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T \right) - \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T \right)^T \tilde{\mathbf{Y}} - \tilde{\mathbf{Y}}^T \left(\overset{\Delta}{\mathbf{X}} \mathbf{w}^T \right) + \tilde{\mathbf{Y}}^T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

becomes

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

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

Finding a local minimum



- \mathbf{w}_0 is local minimum iff for all \mathbf{w} in a neighborhood of \mathbf{w}_0 , $J(\mathbf{w}_0) \leq J(\mathbf{w})$
- If \mathbf{w} is a local minimum then
$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0$$
- \mathbf{w}^* is a global minimum iff
$$\forall \mathbf{w}, J(\mathbf{w}^*) \leq J(\mathbf{w})$$
- 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.

$$\frac{\partial J(\mathbf{w})}{\partial [w_1, w_2, \dots, w_{F+1}]} = \begin{bmatrix} \frac{\partial J(\mathbf{w})}{\partial w_1} \\ \frac{\partial J(\mathbf{w})}{\partial w_2} \\ \vdots \\ \frac{\partial J(\mathbf{w})}{\partial w_{F+1}} \end{bmatrix} \quad \frac{\partial J(\mathbf{w})}{\partial \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{F+1} \end{bmatrix}} = \left[\frac{\partial J(\mathbf{w})}{\partial w_1}, \dots, \frac{\partial J(\mathbf{w})}{\partial w_{F+1}} \right]$$

Partial derivative: formulas

Notations

\mathbf{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:
 $A^T = A$

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

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

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

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

Derivative of J

Cost function

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

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

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

Cancellation of the derivative

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

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

Exercise 27

We consider once again exercise 25 to solve without using the trick of zeroing J which usually does not work.

$$x_{11} = 2 \quad x_{12} = 0.5 \quad y_1 = 1$$

$$x_{21} = 1 \quad x_{22} = 2 \quad y_2 = 0$$

$$x_{31} = 0 \quad x_{32} = 0 \quad y_3 = 1$$

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

$$f_{\mathbf{a},b}(\mathbf{x}) = 1(\mathbf{a} \cdot \mathbf{x} \leq b)$$

with $\mathbf{a} = [a_1, a_2]$. We consider an L2-loss function

$$J(a_1, a_2, b) = L(\mathcal{S}_2, f_{\mathbf{a},b}) = \frac{1}{2} \sum_{n=1}^N (f^v(\mathbf{x}_n) - \tilde{y}_n)^2$$

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

Exercise

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

$$\min_{\mathbf{w}} J(\mathbf{w}) = \tilde{Y}^T \tilde{Y} - \tilde{Y}^T \mathbf{X}^{\Delta} \left(\begin{pmatrix} \Delta^T & \Delta \end{pmatrix} \begin{pmatrix} \mathbf{X} & \mathbf{X} \end{pmatrix} \right)^{-1} \mathbf{X}^{\Delta T} \tilde{Y}$$

Answer to exercise 27 I

① $\mathbf{w} = [-a_1, -a_2, b]$ and $\hat{\mathbf{x}} = [x_1, x_2, 1]$ because

$$f^v(\mathbf{w}) = b - \mathbf{a} \cdot \mathbf{x} = \mathbf{w} \cdot \hat{\mathbf{x}}$$

②

$$\mathbf{X} = \begin{bmatrix} 2 & 0.5 \\ 1 & 2 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{X}} = \begin{bmatrix} 2 & 0.5 & 1 \\ 1 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\hat{\mathbf{X}}^T = \begin{bmatrix} 2 & 1 & 0 \\ 0.5 & 2 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{X}}^T \hat{\mathbf{X}} = \begin{bmatrix} \mathbf{5} & 3 & 3 \\ 3 & \frac{17}{4} & \frac{5}{2} \\ 3 & \frac{5}{2} & 3 \end{bmatrix}$$

$$\mathbf{5} = 2 \times 2 + 1 \times 1 + 0 \times 0$$

③

$$Y = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \tilde{Y} = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{X}}^T \tilde{Y} = \begin{bmatrix} 1 \\ -\frac{3}{2} \\ 1 \end{bmatrix}$$

Answer to exercise 27 II

- 4 Knowing that

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

and based on the solution found in exercise 25, we select $\mathbf{w}^* = [\frac{2}{7}, -\frac{8}{7}, 1]$.

$$\begin{pmatrix} \Delta^T \Delta \\ \mathbf{X} \mathbf{X} \end{pmatrix} \mathbf{w}^{*T} = \begin{bmatrix} 1 \\ -\frac{3}{2} \\ 1 \end{bmatrix} = \begin{matrix} \Delta^T \\ \mathbf{X} \end{matrix} \tilde{\mathbf{Y}}$$

- 5 We consider four samples.

- The size of \mathbf{Y} and $\tilde{\mathbf{Y}}$ is 4×1 .
- The size of \mathbf{X} is 4×2 .
- The size of $\begin{matrix} \Delta \\ \mathbf{X} \end{matrix}$ is 4×3 .

The remaining sizes are unchanged.

- The size of $\begin{matrix} \Delta^T \Delta \\ \mathbf{X} \mathbf{X} \end{matrix}$ and $\begin{pmatrix} \Delta^T \Delta \\ \mathbf{X} \mathbf{X} \end{pmatrix}^{-1}$ is 3×3 .

Answer to exercise 27 III

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

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

$$\begin{aligned}(\mathbf{w}^*)^T &= \left(\begin{array}{cc} \Delta^T & \Delta \\ \mathbf{X} & \mathbf{X} \end{array} \right)^{-1} \mathbf{X}^{\Delta T} \tilde{\mathbf{Y}} \\ \mathbf{w}^* &= \tilde{\mathbf{Y}}^T \mathbf{X}^{\Delta} \left(\begin{array}{cc} \Delta^T & \Delta \\ \mathbf{X} & \mathbf{X} \end{array} \right)^{-1}\end{aligned}$$

We plug this in the definition of J.

$$\begin{aligned}J(\mathbf{w}^*) &= \tilde{\mathbf{Y}}^T \mathbf{X}^{\Delta} \left(\begin{array}{cc} \Delta^T & \Delta \\ \mathbf{X} & \mathbf{X} \end{array} \right)^{-1} \mathbf{X}^{\Delta T} \mathbf{X}^{\Delta} \left(\begin{array}{cc} \Delta^T & \Delta \\ \mathbf{X} & \mathbf{X} \end{array} \right)^{-1} \mathbf{X}^{\Delta T} \tilde{\mathbf{Y}} \\ &\quad - 2 \tilde{\mathbf{Y}}^T \mathbf{X}^{\Delta} \left(\begin{array}{cc} \Delta^T & \Delta \\ \mathbf{X} & \mathbf{X} \end{array} \right)^{-1} \mathbf{X}^{\Delta T} \tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^T \tilde{\mathbf{Y}}\end{aligned}$$

After simplification we get the expected result.

$$J(\mathbf{w}^*) = \tilde{\mathbf{Y}}^T \tilde{\mathbf{Y}} - \tilde{\mathbf{Y}}^T \hat{\mathbf{X}} \left(\hat{\mathbf{X}}^T \hat{\mathbf{X}} \right)^{-1} \hat{\mathbf{X}}^T \tilde{\mathbf{Y}}$$

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.

New notations

- Extended vector $\hat{\mathbf{x}}$
- Extended matrix $\hat{\mathbf{X}}$
- Unique vector \mathbf{w} for linear classifier instead of $[-\mathbf{a}, b]$.
- Derivation w.r. to a row vector $\frac{\partial}{\mathbf{w}}$ or a column vector $\frac{\partial}{\mathbf{w}^T}$.
- \mathbf{w}^* global minimum of the loss function.

Conclusion of subsection 3, 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 $\hat{\mathbf{X}}$.
- Parameter values are obtained by minimizing $\hat{\mathbf{X}}^T \hat{\mathbf{X}}$.

These are techniques requiring the knowledge of Y

In the next section we discuss technique not needing Y .

Table of Contents I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

Table of Contents II

8. Supplementary material regarding matrices

Content of section 4, Predicting the learning performances and probabilistic framework I

- 4.1 Inference on an example
- 4.2 Linear discriminant analysis
- 4.3 Predicting the true probabilities
- 4.4 Prior and Bayes formula

A linear classifier separating gaussians

Exercise 28

Let \mathcal{Y} be a uniform binary random variable and X when conditioned to \mathcal{Y} be a 2D-gaussian variable with mean $\mu_0 \in \mathbb{R}^2$ or $\mu_1 \in \mathbb{R}^2$ and standard deviation $\sigma_0 > 0$ or $\sigma_1 > 0$.

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

$$f_{X|\mathcal{Y}=1}(\mathbf{x}) \geq f_{X|\mathcal{Y}=0}(\mathbf{x}) \Leftrightarrow (\mu_1 - \mu_0)\mathbf{x}^T \geq (\mu_1 - \mu_0)\left(\frac{1}{2}\mu_1 + \frac{1}{2}\mu_0\right)^T$$

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

Answer to exercise 28

1

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

2

$$f_{X|\mathcal{Y}=0}(\mathbf{x}) = \frac{1}{2\pi\sigma_0^2} e^{-\frac{1}{2\sigma_0^2}(\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^\top}$$

$$f_{X|\mathcal{Y}=1}(\mathbf{x}) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{1}{2\sigma_1^2}(\mathbf{x}-\boldsymbol{\mu}_1)(\mathbf{x}-\boldsymbol{\mu}_1)^\top}$$

3

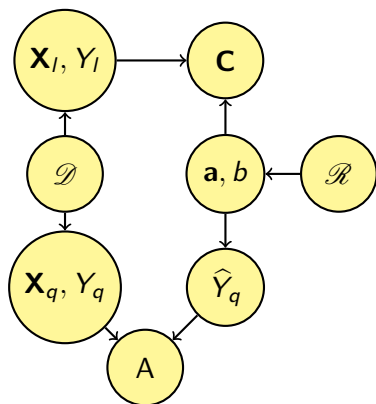
$$\frac{f_{X|\mathcal{Y}=1}(\mathbf{x})}{f_{X|\mathcal{Y}=0}(\mathbf{x})} = e^{\frac{1}{2\pi\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^\top - \frac{1}{2\pi\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_1)(\mathbf{x}-\boldsymbol{\mu}_1)^\top} \geq 1$$

$$\Leftrightarrow (\mathbf{x} - \boldsymbol{\mu}_0)(\mathbf{x} - \boldsymbol{\mu}_0)^\top \geq (\mathbf{x} - \boldsymbol{\mu}_1)(\mathbf{x} - \boldsymbol{\mu}_1)^\top$$

$$\Leftrightarrow -2\boldsymbol{\mu}_0\mathbf{x}^\top + \boldsymbol{\mu}_0\boldsymbol{\mu}_0^\top \geq -2\boldsymbol{\mu}_1\mathbf{x}^\top + \boldsymbol{\mu}_1\boldsymbol{\mu}_1^\top$$

$$\Leftrightarrow (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\mathbf{x}^\top \geq (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\left(\frac{1}{2}\boldsymbol{\mu}_0 + \frac{1}{2}\boldsymbol{\mu}_1\right)^\top$$

An experiment



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

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

Training set: 12 samples

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

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

Confusion matrix

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

Testing set: 2 samples

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

Accuracy: 3 possible values

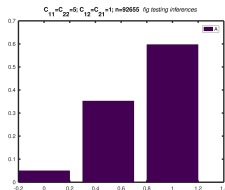
$$A = \frac{1}{2}1(y_{q0} = \hat{y}_{q0}) + \frac{1}{2}1(y_{q1} = \hat{y}_{q1})$$

Algorithm of a random classifier

Require: \mathbf{C}

Ensure: $P(A)$

- 1: Set $P(A) = [0, 0, 0]$.
- 2: **for** $i = 1 : I$ **do**
- 3: **repeat**
- 4: Draw $\mu_0, \mu_1, \sigma_0, \sigma_1, \mathbf{a}$ and b .
- 5: Set $Y_I^T = [0 \dots 0, 1 \dots 1]$.
- 6: Draw \mathbf{X}_I .
- 7: Compute \hat{Y}_I with \mathbf{X}_I and $\hat{\mathbf{C}}$ with Y_I, \hat{Y}_I .
- 8: **until** $\hat{\mathbf{C}} = \mathbf{C}$
- 9: Set $Y_q^T = [0, 1]$.
- 10: Draw \mathbf{X}_q .
- 11: Compute \hat{Y}
- 12: Compute $A = \frac{1}{2}1(\hat{y}_{q0} = 0) + \frac{1}{2}1(\hat{y}_{q1} = 1)$
- 13: Adapt $P(A)$ with A
- 14: Normalize $P(A)$



Conditional probabilities

$$P(A = 0 | \hat{\mathbf{C}} = \mathbf{C}),$$

$$P(A = 0.5 | \hat{\mathbf{C}} = \mathbf{C}),$$

$$P(A = 1 | \hat{\mathbf{C}} = \mathbf{C})$$

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

- $P(A = 1 \text{ and } \widehat{C} = C)$ means the probability of having $A = 1$ **and** that $\widehat{C} = C$
- The assumption implies $P(A = 1 \text{ and } \widehat{C} = C)$ is small.
- If each time $\widehat{C} = C$, we also have $A = 1$ then the assumption makes it invisible in $P(A = 1 \text{ and } \widehat{C} = C)$
- $P(A = 1 | \widehat{C} = C)$ means the probability of having $A = 1$ **given** that $\widehat{C} = C$
- The assumption does not imply anything on $P(A = 1 | \widehat{C} = C)$
- If each time $\widehat{C} = C$, we also have $A = 1$ then $P(A = 1 | \widehat{C} = C) = 1$ 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,
```

- X is here a random **vector**.
- $\mathcal{P}(\dots \& \dots)$ joint probability
- $\mathcal{P}(\dots | \dots)$ conditional probability

Conclusion of subsection 1, 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 4, Predicting the learning performances and probabilistic framework I

- 4.1 Inference on an example
- 4.2 Linear discriminant analysis**
- 4.3 Predicting the true probabilities
- 4.4 Prior and Bayes formula

Exercise 29

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

$$P(y = 0) = P(y = 1) = 0.5 \text{ and } \begin{cases} f_{\mathbf{x}|y=0}(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_0)(\mathbf{x}-\boldsymbol{\mu}_0)^T} \\ f_{\mathbf{x}|y=1}(\mathbf{x}) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(\mathbf{x}-\boldsymbol{\mu}_1)(\mathbf{x}-\boldsymbol{\mu}_1)^T} \end{cases}$$

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.

- 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

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

Answer to exercise 29 II

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

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

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

$$\int_{x_1=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_1-\mu_{01})^2} \int_{x_2=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_2-\mu_{02})^2} =$$
$$\int_{x_1=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_1-\mu_{11})^2} \int_{x_2=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^2}(x_2-\mu_{12})^2} = 1$$

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 29 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 $f_{\mathbf{x}|y=1}(\mathbf{x}) > f_{\mathbf{x}|y=0}(\mathbf{x})$.

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

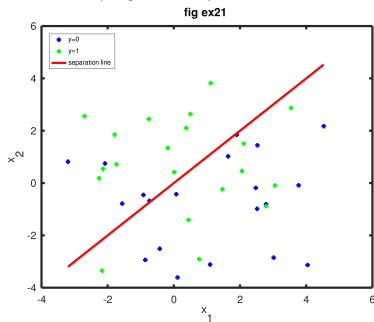
This proves $y = 1$ is more likely when

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

Answer to exercise 29 IV

- 3 $y = 1$ is more likely when $x_2 \geq x_1$. Indeed

$$(\mu_0 - \mu_1)\mathbf{x}^T = x_1 - x_2 \text{ and } \mu_0\mu_0^T - \mu_1\mu_1^T = 0$$



Probabilistic assumption

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

$$f_{\mathbf{x}'|Y=1}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{F}{2}} |\det(\Sigma)|^{\frac{F}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu_1)\Sigma^{-1}(\mathbf{x}-\mu_1)^T}$$
$$f_{\mathbf{x}'|Y=0}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{F}{2}} |\det(\Sigma)|^{\frac{F}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu_0)\Sigma^{-1}(\mathbf{x}-\mu_0)^T}$$

Σ is defined as the covariance matrix

$$\Sigma = \mathcal{E} \left[(\mathbf{x}')^T \mathbf{x}' \right]$$

where \mathcal{E} is the expectation and here \mathbf{x}' is a random row vector.

Covariance matrix

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

$$\hat{\Sigma} = \sum_{n=0}^{N-1} \mathbf{x}_n^T \mathbf{x}_n = \mathbf{X}^T \mathbf{X}$$

Note the striking similarity of this covariance matrix with $\mathbf{X}^{\Delta T} \mathbf{X}^{\Delta}$ 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 29, we compute the logarithm of the ratio of

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

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

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

Supervised feature extraction

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

New notations

- \mathbf{x}^r is a random vector.
- $f_{\mathbf{x}}^r(\mathbf{x})$ is the probability density of $\mathcal{P}(\mathbf{x}^r)$.
- $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ are the mean of the probability distributions of classes 0 and 1. They are estimated using averaging operators on the training set. Their estimates is $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\mu}}_1$.
- $\boldsymbol{\Sigma}$ is the common covariance matrix of both Gaussian probability distributions. It is estimated using the whole training set. Its estimation is denoted $\hat{\boldsymbol{\Sigma}}$.
- $\det(\mathbf{A})$ is the determinant of \mathbf{A} , it is a scalar.

Conclusion of subsection 2, Linear discriminant analysis

When comparing with the L_2 -linear classifier.

- 1 We also have to inverse the covariance matrix.
- 2 Instead of considering the cross-covariance matrix $\mathbf{X}^T \mathbf{Y}$, 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 4, Predicting the learning performances and probabilistic framework I

- 4.1 Inference on an example
- 4.2 Linear discriminant analysis
- 4.3 Predicting the true probabilities**
- 4.4 Prior and Bayes formula

Making inference on hidden parameters based on some evidence

It is common to compute the probability of having a 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.

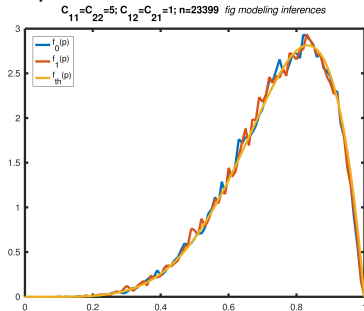
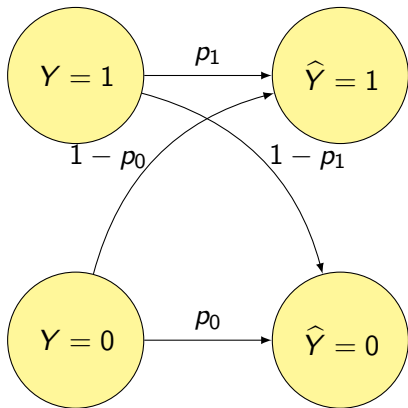
- Given a dataset drawn from a **unique** probability distribution
- Given a classifier drawn from a **unique** probability distribution
- What is the likely accuracy given the confusion matrix computed on a small example of 12 samples.

Beware

This section is meant only to better understand the Bayes formula

Modeling the statistical inference

Here we do not consider the classification problem.



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

Exercise 30

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 \hat{Y} consistent with the confusion matrix is

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

① **Require:** \mathbf{C}, Q, l

Ensure: \mathbf{p}, f_0, 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 \hat{Y} along p_0 and p_1 .

5: Compute $\hat{\mathbf{C}}$ according to \hat{Y} and Y .

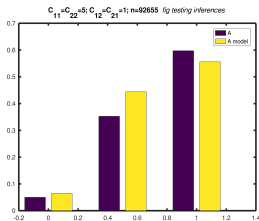
6: **if** $\hat{\mathbf{C}} = \mathbf{C}$ **then**

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.

$P(A|\hat{C} = C)$ 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 random datasets and classification predictors, it measures $P(A|\hat{C} = C)$ 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 $P(A|\hat{C} = C)$.

- $A \rightarrow B$: means that

$$\mathcal{P}(A, B) = \mathcal{P}(B|A)\mathcal{P}(A)$$

- $\binom{n}{p}$ means the number of different subsets of size p that can be drawn out of a set of size n .

$$\binom{n}{p} = \frac{n!}{p!(n-p)!}$$

Conclusion of subsection 3, 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 yields 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 4, Predicting the learning performances and probabilistic framework I

- 4.1 Inference on an example
- 4.2 Linear discriminant analysis
- 4.3 Predicting the true probabilities
- 4.4 Prior and Bayes formula

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** $i = 1 : I$ **do**

2: Draw $\mu_0, \mu_1, \sigma_0, \sigma_1, \mathbf{a}$ and b .

3: Set $Y_i^T = [0 \dots 0, 1 \dots 1]$.

4: Draw \mathbf{X}_i .

5: Compute \hat{Y}_i with \mathbf{X}_i

6: Compute p_0 and p_1 by comparing \hat{Y}_i and Y_i .

Do we need a prior to compute a conditional probability?

No

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

Yes

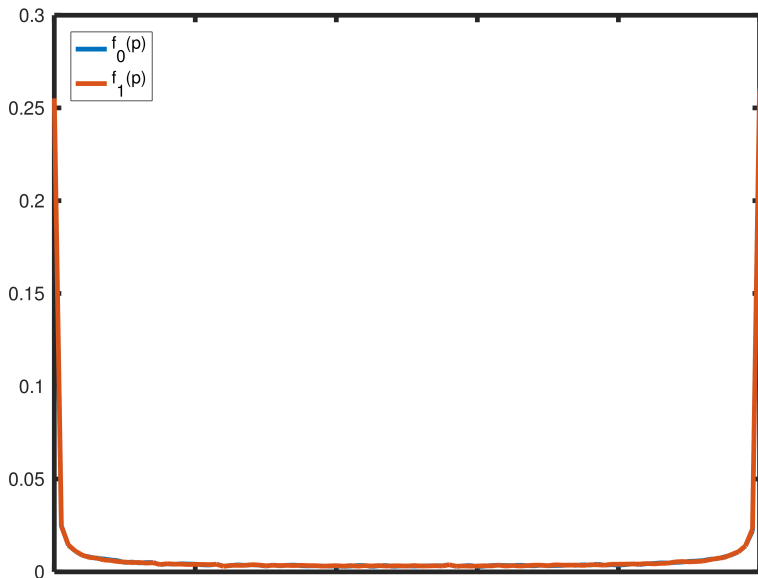
To compute $\mathcal{P}(p_0, p_1|\mathbf{C})$ we consider all possible values of p_0 and p_1 and for each compute a probability of $\mathcal{P}(\mathbf{C}|p_0, p_1)$ and by counting the number of draws for which C has the appropriate value we get a probability of p_0, p_1 . But the relative importance of p_0, p_1 is precisely a **prior**. In exercise 30, p_0 and p_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 $\mathcal{P}(p_0, p_1|\mathbf{C})$ and $\mathcal{P}(\mathbf{C}|p_0, p_1)$ are proportionate is actually a choice of prior that might be a not too bad choice.

Measured prior for this very specific problem

Prior n=148223 *fig modeling prior*



Bayes formula

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

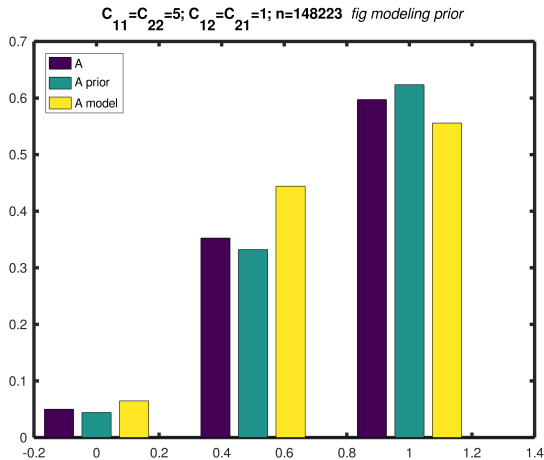
Applying this formula in our context

$$f_{A|\hat{\mathbf{C}}}(a, \mathbf{C}) = \int_{p_0, p_1} f_{A|\hat{\mathbf{C}}, p_0, p_1}(a, \mathbf{C}, p_0, p_1) f_0(p_0) f_1(p_1) dp_0 dp_1$$

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

This posterior probability distribution of 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.

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

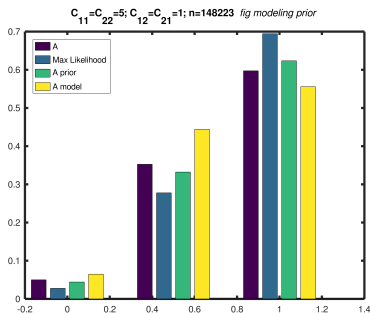
$$\text{Since } \frac{d}{dp} p^5(1-p) = 0 \Rightarrow 5 - 6p = 0 \Rightarrow p = \frac{5}{6}$$

We then get the distribution of A

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

$$\mathcal{P}(A|\hat{\mathbf{C}} = \mathbf{C}, p_0 = p_1 = \frac{5}{6})$$

This new distribution of A is shown in blue.



Conclusion

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

New notations

- $\neg A$ is the alternative event to A .

Conclusion of section 4, 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 I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

Table of Contents II

8. Supplementary material regarding matrices

Content of section 5, More in depth with probabilities I

5.1 Probabilities

5.2 Using Gaussians

5.3 Probabilities as a loss function designer

Histograms are empirical approximation of probability distributions I

- Let $I(x, y)$ be a continuous image $I(x, y)$ (i.e. an image defined with real-valued coordinates)
- Let X and Y be independent uniform random variables
- The histogram of I is an approximation of the Z -probability distribution

$$Z = I(X, Y)$$

- The probability distribution is the derivative of the cumulative distribution (for sufficiently regular random variables)

$$f_Z(z) = \frac{d}{dz} \mathcal{P}(Z \leq z)$$

Example of a continuous image

Exercise 31

We consider here a continuous image.

$$I(x, y) = x^2 1(x \in [-1, 1]) 1(y \in [-1, 1])$$

Let Z be the random variable yielding the value of $I(x, y)$ when a point is selected randomly in the image.

- 1 Prove that

$$\mathcal{P}(Z \leq z) = \frac{1}{4} \int_{x=-1}^1 \int_{y=-1}^1 1(x^2 \leq z) dx dy$$

- 2 Show that

$$\mathcal{P}(Z \leq z) = \sqrt{z} 1(z \in [0, 1]) + 1(z > 1)$$

- 3 Show that

$$f_Z(z) = \frac{1}{2} \frac{1}{\sqrt{z}} 1(z \in [0, 1])$$

- 4 Write the code to check this last statement.

- ① The probability distribution of the uniform law on $[-1, 1] \times [-1, 1]$ is

$$f_U(x, y) = 1(x \in [-1, 1])1(y \in [-1, 1])\frac{1}{4}$$

$$\begin{aligned}\mathcal{P}_Z(Z \leq z) &= \mathcal{P}_{X,Y}(\mathcal{I}(X, Y) \leq z) = \int_x \int_y 1(\mathcal{I}(x, y) \leq z) f_U(x, y) dx dy \\ &= \frac{1}{4} \int_{x=-1}^1 \int_{y=-1}^1 1(x^2 \leq z) dx dy\end{aligned}$$

- ② Let us assume $z \in [0, 1]$.

$$\begin{aligned}\mathcal{P}_Z(Z \leq z) &= \frac{2}{4} \int_{-1}^1 1(x^2 \leq z) dx \int_0^1 1(x^2 \leq z) dx \\ &= \int_0^1 1(x \leq \sqrt{z}) dx = \int_0^{\sqrt{z}} dx = \sqrt{z}\end{aligned}$$

Let us assume $z < 0$, $\mathcal{P}(Z \leq z) = \mathcal{P}(Z < 0) = 0$

Let us assume $z > 1$, $\mathcal{P}(Z \leq z) \geq \mathcal{P}(Z \geq 1) = 1$

Therefore

$$\mathcal{P}(Z \leq z) = \sqrt{z}1(z \in [0, 1]) + 1(z > 1)$$

Answer to exercise 31 II

3

$$f_Z(z) = \frac{\partial}{\partial z} \mathcal{P}(Z \leq z) = \frac{1}{2\sqrt{z}} 1(z \in [0, 1])$$

- 4 The following algorithm approximates $f_Z(z)$ ($\mathcal{U}(-1, 1)$ is the uniform law on $[-1, 1]$).

Require: z

Ensure: $f_Z(z)$

- 1: Set $N = 10^7$, $h = 0.01$
- 2: Draw randomly a vector X of size $N \times 1$ whose components sample $\mathcal{U}(-1, 1)$
- 3: Count n the number of components of X that fulfill $x_n^2 \in [z, z + h)$.
- 4: Yield $\frac{n}{hN}$

This algorithm can be tested several times by drawing randomly z and checking that the difference with $f_Z(z)$ remains small.

- The mean value of an image is

$$\mathcal{E}[Z] = \frac{\int_x \int_y I(x, y) dx dy}{\int_x \int_y 1(x, y \in \mathcal{I}) dx dy}$$

- The k -th q -quantile is

$$\mathcal{P}[Z \leq z_{\text{quantile}}] = kq \quad \Rightarrow \quad z_{\text{quantile}} = (z \mapsto \mathcal{P}[Z \leq z])^{-1}(kq)$$

Example on the continuous image

Exercise 32

We consider again the following continuous image.

$$I(x, y) = x^2 1(x \in [-1, 1]) 1(y \in [-1, 1])$$

Let Z be the random variable yielding the value of $I(x, y)$ when a point is selected randomly in the image.

- 1 Using Z , show that $\mathcal{E}Z = \frac{1}{3}$. And show that the mean value of I is also $\frac{1}{3}$.
- 2 Compute the first and third quartile using the two techniques. The result is $\frac{1}{16}$ and $\frac{9}{16}$.

1

$$EZ = \int_{z=0}^1 z \frac{1}{2} \frac{1}{\sqrt{z}} dz = \frac{1}{2} \int_{z=0}^1 \sqrt{z} dz = \frac{1}{2} \left[\frac{2}{3} z^{3/2} \right]_0^1 = \frac{1}{3}$$

The mean of the image is

$$\frac{1}{4} \int_{x=-1}^1 \int_{y=-1}^1 x^2 dx dy = \frac{1}{2} \int_{x=-1}^1 x^2 dx = \frac{1}{2} \left[\frac{1}{3} x^3 \right]_{-1}^1 = \frac{1}{3}$$

Chebychev Inequality

For X a Gaussian random variable

$$\mathcal{P}(|X - \mathcal{E}X| \geq c) = 1 - \operatorname{erf}\left(\frac{c}{\sqrt{2\operatorname{Var}X}}\right) \quad (5)$$

where $\operatorname{erf}(x)$ is defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

When X is not necessarily a Gaussian random variable,

$$\mathcal{P}(|X - \mathcal{E}X| \geq c) \leq \frac{\operatorname{Var}X}{c^2} \quad (6)$$

For sufficiently regular random variables, the probability distribution is related to the cumulative distribution

$$f_Z(z) = \frac{d}{dz} \mathcal{P}(Z \leq z)$$

Exercise 33

Software generally include the erf function, however to save time, it can be useful to have a quick way to approximate it.

- 1 *Using its integral formula, write a formula to approximate it.*
- 2 *Consider a Gaussian random variable on mean 0 and standard deviation 1, and check equation (5) so as to give a high level of confidence in this equation.*

Answer 1/2 to exercise 33

- ① The integral is approximated with N rectangles of width $\frac{|x|}{N}$ and height $e^{-\frac{t^2}{2}}$ paving the $[0, |x|]$. Note that the erf function is odd.

$$y = \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \approx \frac{2}{\sqrt{\pi}} \frac{x}{N} \sum_{k=0}^{N-1} e^{-\frac{k^2 x^2}{N^2}}$$

Require: $x, N,$

Ensure: y

1: $y = 0$

2: **for** $k = 0 : N - 1$ **do**

3: Add to $y, e^{-\frac{k^2 x^2}{N^2}}$

4: Multiply y with $\frac{2}{\sqrt{\pi}} \frac{x}{N}$

```
>> c = -7.7189
```

```
>> y_app = -1.0044
```

```
>> y_th = -1
```

```
>> y_app-y_th= -4.3549e-03
```

Answer 2/2 to exercise 33

- 2 Generate 1000 random numbers for X , denoted x_i for $i \in \{1 \dots I\}$. An approximation of the left part of (5) is

$$\frac{1}{I} \sum_{i=1}^I 1(|x_i| \leq c)$$

```
xi=randn(1,1000);  
c=2*rand(1);  
p_app=mean(abs(xi)>=c);  
p_th=1-erf(c/sqrt(2));  
c,p_app,p_th,  
>>c = 1.1490  
>> p_app = 0.2510  
>> p_th = 0.2505
```

Transforming a graph into an empirical distribution

Require: (x_n, y_n) and K

Ensure: (x'_k, n_k)

1: Compute the ranging interval

$$a = \min_n x_n \text{ and } b = \max_n x_n$$

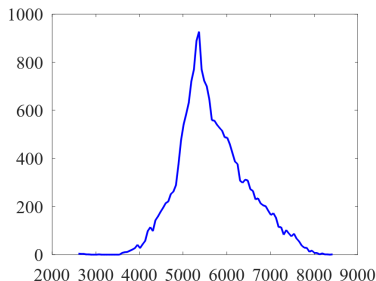
2: **for** $k = 0 : K - 1$ **do**

$$3: \quad x'_k = a + k \frac{b-a}{K-1}$$

$$4: \quad n_k = \sum_{n=0}^{N-1} 1(x_n \in [x_k, x_{k+1}))$$

Generally, K is chosen

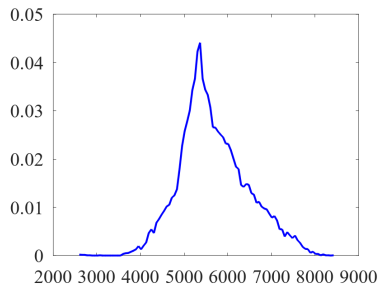
$$K \approx \sqrt{N}$$



Normalizing empirical distributions

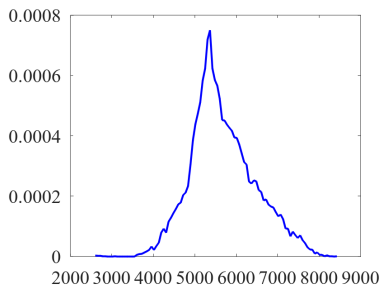
Classical technique

$$n'_k = \frac{n_k}{\sum_{k=0}^{K-1} n_k}$$



Technique required to compare with parametric distributions

$$n'_k = \frac{n_k}{\sum_{k=0}^{K-1} n_k (x_{k+1} - x_k)}$$



Exercise 34

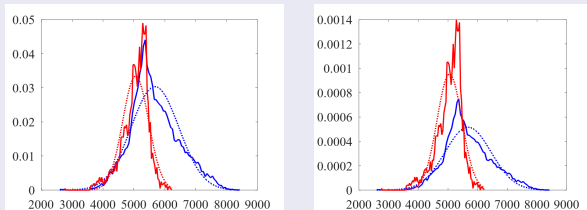


Figure 6: Empirical distributions of the bandwidth number 50 considering all pixels in blue and only pixels showing soybean in red. The dotted curves are the approximate Gaussian distributions.

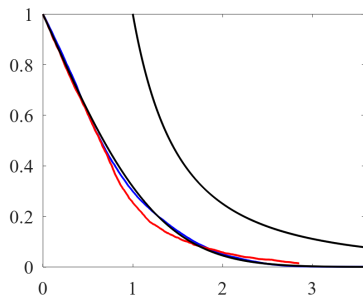
- 1 Write the pseudo-code of an algorithm yielding figure 6, empirical distributions are such that their sums equal 1.
- 2 Write the pseudo-code of an algorithm yielding figure 6, empirical distributions are such that their approximate integral equal 1.

Exercise 35

Let us call X and Y two empirical distributions obtained using the intensity values at bandwidth number 50 and conditionally to being actually soybean (i.e. labels 10, 11, 12 of the groundtruth map).

- 1 Transform X and Y into centered and normalized random variables denoted X_r and Y_r .
- 2 Plot as a function of $c \in [0, 2]$ the left side of equation (6) for X_r and Y_r . Plot the right side of equation (6) and that of equation (5).

Answer 1/2 to exercise 35



- The lower curve is for Gaussian random variable.
- The blue curve is obtained using the hyperspectral image at bandwidth number 50.
- The red curve is obtained using the hyperspectral image at bandwidth number 50 considering only the pixels where land is covered with soybeans.
- The upper black curve is the Chebychev upperbound.

Answer 2/2 to exercise 35

- 1 First stack in \mathbf{x} a column vector the intensities at the bandwidth number 50.

$$\mathbf{x}_r = \frac{\mathbf{1}}{\sigma}(\mathbf{x} - \mu\mathbf{1}) \quad \text{where} \quad \begin{cases} \sigma = \sqrt{\frac{1}{N} \sum_{n=1}^N (x_n - \mu)^2} \\ \mu = \frac{1}{N} \sum_{n=1}^N x_n \end{cases}$$

- 2 **Require:** \mathbf{x} , K

Ensure: (C, P)

1: $y = 0$,

2: **for** $k = 0 : K$ **do**

3: $C_k = \frac{k}{K}$

4: $P_k = \frac{1}{N} \sum_{n=1}^N \mathbf{1}(|x_n| \geq C_k)$

Proving the Chebychev inequality

Exercise 36

A simple proof of the Chebychev arises from the following steps.

- 1 *Prove the Markov inequality which states*

$$\mathcal{P}[|X| \geq c] \leq \frac{\mathcal{E}|X|}{c}$$

To do so, introduce a new random variable $Y = c1(|X| \geq c)$ and show that it is upper bounded by X and compute its expectancy.

- 2 *Apply the Markov inequality to $Z = (X - \mathcal{E}X)^2$.*

Answer 1/2 to exercise 36

- 1 Let Y be a new random variable

$$Y = c1(|X| \geq c)$$

We first prove that $Y \leq |X|$

$$\begin{cases} Y \leq c \leq |X| & \text{if } |X| \geq c \\ Y \leq 0 \leq |X| & \text{if } |X| < c \end{cases}$$

This proves that $\mathcal{E}[Y] \geq |X|$

We then compute $\mathcal{E}[Y]$

$$\mathcal{E}[Y] = c\mathcal{P}(Y = c) + 0\mathcal{P}(Y = 0) = c\mathcal{P}(|X| \geq c)$$

- 2 To prove equation (6), $\mathcal{P}(|X - \mathcal{E}X| \geq c) \leq \frac{\mathcal{V}arX}{c^2}$ we apply the Markov inequality

$$\mathcal{P}(Z \geq c^2) \leq \frac{\mathcal{E}[Z]}{c^2} = \frac{\mathcal{V}ar(X)}{c^2}$$

- erf is the error function (a.k.a. Gauss error function).

Conclusion of subsection 1, Probabilities

Content of section 5, More in depth with probabilities I

5.1 Probabilities

5.2 Using Gaussians

5.3 Probabilities as a loss function designer

Thresholding using Gaussian probability distributions I

Given a set of intensities I_n , we model the membership with each cluster as

$$f_{I_n|n \in \mathcal{N}_0, \mu_0, \sigma_0}(I_n) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(I_n - \mu_0)^2}{2\sigma_0^2}} = g_{\mu_0, \sigma_0}(I_n)$$

$$f_{I_n|n \in \mathcal{N}_1, \mu_1, \sigma_1}(I_n) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(I_n - \mu_1)^2}{2\sigma_1^2}} = g_{\mu_1, \sigma_1}(I_n)$$

seg_thresholding3.m

Require: I

Ensure: T

- 1: select an initial value for T ,
- 2: **while** T is modified **do**
- 3: $\mathcal{N}_0 = \{n \in \mathcal{N} \mid f_{I_n|n \in \mathcal{N}_0, \mu_0, \sigma_0}(I_n) > f_{I_n|n \in \mathcal{N}_1, \mu_1, \sigma_1}(I_n)\}$
- 4: $\mathcal{N}_1 = \mathcal{N} \setminus \mathcal{N}_0$
- 5: $\mu_0, \sigma_0 \in \operatorname{argmax}_{\mu, \sigma} f_{I_n|n \in \mathcal{N}_0, \mu, \sigma}(I_n)$
- 6: $\mu_1, \sigma_1 \in \operatorname{argmax}_{\mu, \sigma} f_{I_n|n \in \mathcal{N}_1, \mu, \sigma}(I_n)$

Thresholding using Gaussian probability distributions II

This another iterated algorithm maximizing the *likelihood* using crisp assignments.

Normalizing the probability density

Note that to answer the third question, one needs a correct normalization $\frac{1}{\sqrt{2\pi}\sigma}$. This can be an issue.

Exercise 37

- 1 Prove that step 3 in `seg_thresholding3.m` is

$$T = \frac{\sigma_1 \mu_0 + \sigma_0 \mu_1}{\sigma_0 + \sigma_1} \text{ and } \mathcal{N}_0 = \{n | I_n \leq T\}$$

- 2 Prove that in steps 5, 6, μ_0 and μ_1 should be the average of samples in \mathcal{N}_0 and \mathcal{N}_1 .
- 3 Prove that in steps 5, 6, σ_0 and σ_1 should be the standard deviation of samples in \mathcal{N}_0 and \mathcal{N}_1 .

- 1 State 3 is

$$\mathcal{N}_0 = \{n \in \mathcal{N} \mid f_{I_n | n \in \mathcal{N}_0, \mu_0, \sigma_0}(I_n) > f_{I_n | n \in \mathcal{N}_1, \mu_1, \sigma_1}(I_n)\}$$

where $f_{I_n | n \in \mathcal{N}_0, \mu, \sigma}(I_n) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{\sigma^2}}$

Let us assume $\mu_0 < \mu_1$ and let us define $T = \frac{\sigma_1\mu_0 + \sigma_0\mu_1}{\sigma_0 + \sigma_1}$. Because $T \leq \mu_1$, we have

$$I_n \leq T \Leftrightarrow (\sigma_0 + \sigma_1)I_n - (\sigma_1\mu_0 + \sigma_0\mu_1)$$

$$\Leftrightarrow \frac{I_n - \mu_0}{\sigma_0} + \frac{I_n - \mu_1}{\sigma_1} < 0$$

$$\Leftrightarrow \frac{|I_n - \mu_0|}{\sigma_0} < \frac{|I_n - \mu_1|}{\sigma_1}$$

- ② To make it easier, μ_0, μ_1 and σ_0, σ_1 are here replaced with μ and σ

$$f_{r|I|\mu,\sigma}(I) = \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2} \frac{(I_n - \mu)^2}{\sigma^2}}$$
$$-\ln \left(f_{r|I|\mu,\sigma}(I) \right) = N \ln(\sqrt{2\pi}\sigma) + \sum_{n=0}^{N-1} \frac{1}{2} \frac{(I_n - \mu)^2}{\sigma^2}$$

We are looking for $\mu \in \underset{\mu}{\operatorname{argmax}} f_{r|I|\mu,\sigma}(I) = \underset{\mu}{\operatorname{argmax}} -\ln \left(f_{r|I|\mu,\sigma}(I) \right)$

Such values cancel the partial derivative w.r. to μ .

$$\frac{\partial}{\partial \mu} \left[-\ln \left(f_{r|I|\mu,\sigma}(I) \right) \right] = 0 \Leftrightarrow \sum_{n=0}^{N-1} (I_n - \mu) = 0$$
$$\Leftrightarrow \mu = \frac{1}{N} \sum_{n=0}^{N-1} I_n$$

From a mathematical viewpoint, we would need to make sure that this maximizes the probability.

- 3 The only difference is that the partial derivative is with respect to σ .

$$\begin{aligned}\frac{\partial}{\partial \sigma} \left[-\ln \left(f_{I|\mu,\sigma}(I) \right) \right] = 0 &\Leftrightarrow \frac{N}{\sigma} - \sum_{n=0}^{N-1} \frac{(I_n - \mu)^2}{\sigma^3} = 0 \\ &\Leftrightarrow \sigma^2 = \frac{1}{N} \sum_{n=0}^{N-1} (I_n - \mu)^2\end{aligned}$$

Definition

The likelihood is the probability that the observations fit with the model with its parameters.

$$L(I, \Theta)$$

where Θ is the set of parameters.

In exercise 37, we need new parameters $\mathcal{N}_0, \mathcal{N}_1$ to define the crisp assignments:

$$n \in \mathcal{N}_0 \Leftrightarrow I_n \leq T \quad \text{and} \quad n \in \mathcal{N}_1 \Leftrightarrow I_n > T$$

$$L(I, \mu_0, \mu_1, \sigma, \mathcal{N}_0, \mathcal{N}_1) = \prod_{n \in \mathcal{N}_0} \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{1}{2} \frac{(I_n - \mu_0)^2}{\sigma_0^2}} \prod_{n \in \mathcal{N}_1} \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{1}{2} \frac{(I_n - \mu_1)^2}{\sigma_1^2}}$$

Maximizing the likelihood

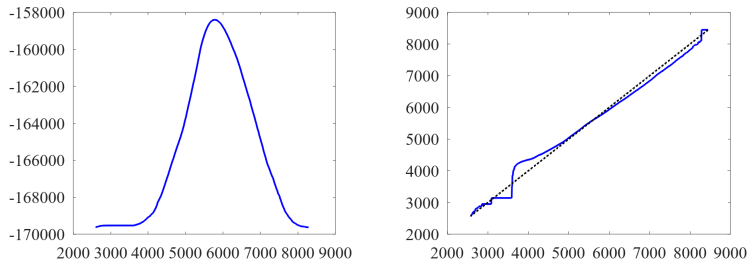


Figure 7: Left: log-likelihood w.r. to T . Right: new threshold T' w.r. to old threshold T .

Exercise 38

- 1 Using right of figure 7, define the catchment areas (a.k.a. basins of attraction): the set of values of T such that the algorithm converges to a given value.

New notations

- $g_{\mu,\sigma}(x)$ is the Gaussian deterministic function.
- L is here the likelihood, it is used as the opposite of a loss function.

Conclusion of subsection 2, Using Gaussians

- Catchment areas in algorithms.
- Likelihood. (discrete or dense probability distribution).

Content of section 5, More in depth with probabilities I

5.1 Probabilities

5.2 Using Gaussians

5.3 Probabilities as a loss function designer

theorem

$$f''(x) > 0 \Rightarrow f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y)$$

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

Recursively or using...

$$f\left(\sum_i \alpha_i x_i\right) \leq \sum_i \alpha_i f(x_i) \text{ when } \sum_i \alpha_i = 1 \text{ and } \forall i, \alpha_i \geq 0$$

First view-point on the Expectation-Minimization algorithm I

- Θ : set of parameters
- $\tilde{\mathbf{x}}$: random process modeling the observations (i.e. intensities)
- \mathbf{x} : observation values (i.e. pixel intensities)
- Y : random variable modeling the hidden states (i.e. labels).
- \mathbf{y} : actual states (i.e. pixel labels), with Ω_y as the set of all possible values
- L : likelihood
- LL : log-likelihood

$$L(\mathbf{x}, \Theta) = \mathcal{P}(X|\Theta) = \sum_{\Omega_y} \mathcal{P}(X, Y = \mathbf{y}|\Theta)$$
$$\sum_{\Omega_y} \mathcal{P}(X|Y = \mathbf{y}, \Theta) \mathcal{P}(Y = \mathbf{y})$$

This is an iterated algorithm and at step $t + 1$, the parameter values $\Theta^{(t)}$ are available.

$$L(\mathbf{x}, \Theta, \Theta^{(t)}) = \sum_{\Omega_y} \mathcal{P}(X|Y = \mathbf{y}, \Theta, \Theta^{(t)}) \mathcal{P}(Y = \mathbf{y}|\Theta^{(t)})$$

First view-point on the Expectation-Minimization algorithm II

We consider the log-likelihood:

$$LL(\mathbf{x}, \Theta, \Theta^{(t)}) = \ln \left(\sum_{\Omega_y} \mathcal{P}(\mathbf{X} | \mathbf{Y} = \mathbf{y}, \Theta, \Theta^{(t)}) \mathcal{P}(\mathbf{Y} = \mathbf{y} | \Theta^{(t)}) \right)$$

Two caveats

$\mathcal{P}(\mathbf{Y} = \mathbf{y} | \Theta^{(t)})$ uses actually the data \mathbf{X} considered as part of $\Theta^{(t)}$.
When the components of \mathbf{X} and \mathbf{Y} are modeled as given- Θ independent random variables, the summation over Ω_y is restricted to each component of \mathbf{y} and \ln is inside a first summation over the components.

The expectation step is

$$y_n^{(t+1)} = \operatorname{argmax}_{y_n} \text{ML}(\Theta^{(t)}, \mathbf{x}, y_0^{(t)} \dots y_n \dots y_{N-1}^{(t)})$$

The maximization step is

$$\Theta^{(t+1)} = \operatorname{argmax}_{\Theta} \text{ML}(\Theta^{(t)}, \mathbf{x}, \mathbf{y}^{(t+1)})$$

First view-point on the Expectation-Minimization algorithm III

Because the logarithm is concave, we get a lower bound called $Q(\Theta, \Theta^{(t)})$

$$LL(\mathbf{x}, \Theta, \Theta^{(t)}) \geq \sum_{\Omega_y} \mathcal{P}(Y = \mathbf{y} | \Theta^{(t)}) \ln \mathcal{P}(X | Y = \mathbf{y}, \Theta, \Theta^{(t)})$$

There are two steps:

- Expectation-step: computing $Q(\Theta, \Theta^{(t)})$. That is fill in the unknown or hidden parameters with most likely possible values computed using observations and previous values of parameters, and weighing these values with their probabilities.
- Maximization-step: finding Θ by maximizing $Q(\Theta, \Theta^{(t)})$.

We get here soft assignments.

Exercise 39

We consider a statistical model for a binary classification problem:

- *The intensity of each pixel follows a Gaussian random variable.*
- *There is a unique standard deviation σ .*
- *The mean value depending on its class membership μ_0, μ_1 .*
- *Conditionally to their classes, the random variables are independent.*

We use this model to infer the parameters' values involved in the model and the hidden parameters by observing only the pixel values.

- 1 *List the variables whose values are known and those whose values are to find by maximizing the likelihood.*
- 2 *Write the likelihood of a given pixel's intensity and that of all N pixels, assuming we know which pixels follows which Gaussian variable.*

ML with equal standard deviation II

- We now assume the assignment of pixels with each probability distribution are N Gaussian probability distributions Y_n .
- The goal is to write the relationship between two successive iterations.
- All parameters have now an indication of the iteration using t as an integer.

$$y_0^{(t)} \cdots y_{N-1}^{(t)}, \mu_0^{(t)}, \mu_1^{(t)}$$

We denote $q_n^{(t)} = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{1}(y_n = 0)$

Exercise 40

We consider the statistical model of exercise 39

- 1 Write the prior probability of Y_n knowing parameters of the last iteration (i.e. t -iteration).*
- 2 Write the posterior probability of Y_n knowing parameters of the last iteration (i.e. t -iteration) and using the pixel intensity values.*
- 3 Write the expectation step of the E-M algorithm, assuming $\mu_1 > \mu_0$.*
- 4 Write the maximization step of the E-M algorithm.*

Answer to exercise 39 I

- 1
 - The parameters whose values are known are those of the observations:
 $N, x_0 \dots x_{N-1}$.
 - The parameters whose values are to be found: μ_0, μ_1, σ .
 - Hidden variables: $y_0 \dots y_{N-1}$.
- 2 For a specific pixel, we have

$$f_{I_n=x|Y_n=0, \Theta^{(t)}}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu_0)^2}{2\sigma^2}}$$

$$f_{I_n=x|Y_n=1, \Theta^{(t)}}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu_1)^2}{2\sigma^2}}$$

The likelihood is

$$L(I, Y, \mu_0, \mu_1, \sigma) = \prod_{n=0}^{N-1} \left(1(y_n = 0) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu_0)^2}{2\sigma^2}} + 1(y_n = 1) \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu_1)^2}{2\sigma^2}} \right)$$

Answer to exercise 39 I

- ① At the last iteration, $y_n^{(t)}$ have received integer values, this is a prior:

$$\mathcal{P}(Y_n = 0 | \Theta^{(t)}) = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{1}(y_n^{(t)} = 0) = q_n^{(t)}$$

$$\mathcal{P}(Y_n = 1 | \Theta^{(t)}) = 1 - \mathcal{P}(Y_n = 0) = 1 - q_n^{(t)}$$

- ② The posterior probability is obtained with the likelihood and the prior:

$$f_{Y_n=y_n | I_n, \Theta^{(t)}}(I_n) = \frac{f_{I_n | Y_n=y_n, \Theta^{(t)}}(I_n) \mathcal{P}(Y_n=y_n | \Theta^{(t)})}{f_{I_n | Y_n=0, \dots}(I_n) \mathcal{P}(Y_n=0 | \dots) + f_{I_n | Y_n=1, \dots}(I_n) \mathcal{P}(Y_n=1 | \dots)}$$

Denoting $q_n^{\prime(t)} = \frac{q_n g_{\mu_0^{(t)}, \sigma}(I_n)}{q_n g_{\mu_0^{(t)}, \sigma}(I_n) + (1 - q_n) g_{\mu_1^{(t)}, \sigma}(I_n)}$ we can write

$$f_{Y_n=y_n | I_n, \Theta^{(t)}} = q_n^{\prime(t)} \mathbf{1}(y_n = 0) + (1 - q_n^{\prime(t)}) \mathbf{1}(y_n = 1)$$

Answer to exercise 39 II

- 3 The objective is to find the parameters maximizing the likelihood of the data.

$$\begin{aligned} \ln f_{Y, I, \Theta | \Theta^{(t)}}^r(I) \\ = \ln \prod_{n=0}^{N-1} \left(f_{I_n, Y_n=0, \Theta | \Theta^{(t)}}^r(I_n) + f_{I_n, Y_n=1, \Theta | \Theta^{(t)}}^r(I_n) \right) \end{aligned}$$

We use the posterior probabilities and the likelihood of I_n to compute these probabilities:

$$\begin{aligned} f_{I_n, Y_n=0, \Theta | \Theta^{(t)}}^r(I_n) &= q_n'^{(t)} f_{I_n, \Theta | Y_n=0, \Theta^{(t)}}^r(I_n) = q_n'^{(t)} g_{\mu_0, \sigma}(I_n) \\ f_{I_n, Y_n=1, \Theta | \Theta^{(t)}}^r &= (1 - q_n'^{(t)}) f_{I_n, \Theta | Y_n=1, \Theta^{(t)}}^r(I_n) \\ &= (1 - q_n'^{(t)}) g_{\mu_1, \sigma}(I_n) \end{aligned}$$

Combining

$$\begin{aligned} \ln \mathcal{P}(Y_n = y_n, I, \Theta | \Theta^{(t)}) &= \\ \ln \prod_{n=0}^{N-1} \left(q_n'^{(t)} g_{\mu_0, \sigma}(I_n) + (1 - q_n'^{(t)}) g_{\mu_1, \sigma}(I_n) \right) \\ &= \sum_{n=0}^{N-1} \ln \left(q_n'^{(t)} g_{\mu_0, \sigma}(I_n) + (1 - q_n'^{(t)}) g_{\mu_1, \sigma}(I_n) \right) \end{aligned}$$

Because of the concavity of \ln , we get a lower bound

$$\begin{aligned} \ln f_{Y_n=y_n, I_n, \Theta|\Theta^{(t)}}(I_n) &\geq \sum_{n=0}^{N-1} q_n'(t) \ln g_{\mu_0, \sigma}(I_n) + (1 - q_n'(t)) \ln g_{\mu_1, \sigma}(I_n) \\ &= -\sum_{n=0}^{N-1} q_n'(t) \frac{(I_n - \mu_0)^2}{2\sigma^2} + (1 - q_n'(t)) \frac{(I_n - \mu_1)^2}{2\sigma^2} \end{aligned}$$

- 4 To approximate $\operatorname{argmax}_{\Theta} Q(\Theta|\Theta^{(t)})$ we maximize the lower bound denoted $Q(\Theta|\Theta^{(t)})$:

$$\begin{aligned} -Q(\Theta|\Theta^{(t)}) &= \sum_{n=0}^{N-1} q_n'(t) \frac{(I_n - \mu_0)^2}{2\sigma^2} + q_n' \ln \sqrt{2\pi}\sigma \\ &\quad + (1 - q_n'(t)) \frac{(I_n - \mu_1)^2}{2\sigma^2} + (1 - q_n') \ln \sqrt{2\pi}\sigma \\ &= N \ln \sqrt{2\pi}\sigma + \sum_{n=0}^{N-1} q_n'(t) \frac{(I_n - \mu_0)^2}{2\sigma^2} + (1 - q_n'(t)) \frac{(I_n - \mu_1)^2}{2\sigma^2} \end{aligned}$$

This maximization is obtained by zeroing the derivatives w.r. to μ_0 , μ_1 and σ .

Answer to exercise 39 IV

- Finding μ_0

$$0 = \frac{\partial}{\partial \mu_0} [-Q(\Theta|\Theta^{(t)})] = -\frac{1}{\sigma^2} \sum_{n=0}^{N-1} q'_n (I_n - \mu_0)$$
$$\Rightarrow \mu_0 = \frac{\sum_{n=0}^{N-1} q'_n I_n}{\sum_{n=0}^{N-1} q'_n}$$

- Finding μ_1

$$0 = \frac{\partial}{\partial \mu_1} [-Q(\Theta|\Theta^{(t)})] = -\frac{1}{\sigma^2} \sum_{n=0}^{N-1} (1 - q'_n) (I_n - \mu_1)$$
$$\Rightarrow \mu_1 = \frac{\sum_{n=0}^{N-1} (1 - q'_n) I_n}{\sum_{n=0}^{N-1} (1 - q'_n)}$$

- Finding σ

$$0 = \frac{\partial}{\partial \sigma} [-Q(\Theta|\Theta^{(t)})] =$$
$$\frac{N}{\sigma} - \frac{1}{\sigma^3} \sum_{n=0}^{N-1} q'_n (I_n - \mu_0)^2 - \frac{1}{\sigma^3} \sum_{n=0}^{N-1} (1 - q'_n) (I_n - \mu_1)^2$$
$$\Rightarrow N\sigma^2 = \sum_{n=0}^{N-1} q'_n (I_n - \mu_0)^2 + \sum_{n=0}^{N-1} (1 - q'_n) (I_n - \mu_1)^2$$

Maximizing the likelihood

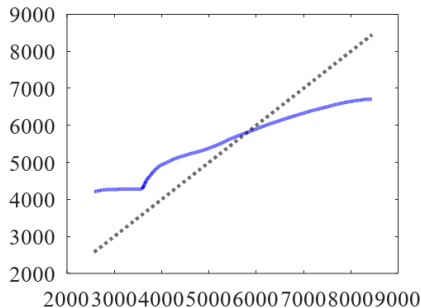
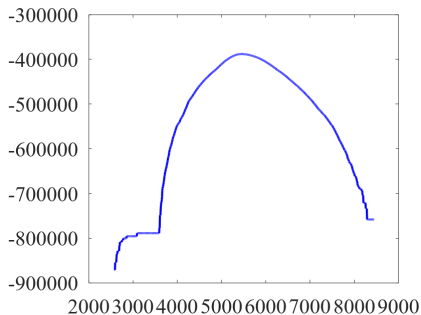


Figure 8: Left: log-likelihood w.r. to T . Right: new threshold T' w.r. to old threshold T .

What could explain the fact that on the left figure, the probability appears more flat than on previous experiments?

Second viewpoint on the Expectation-Minimization algorithm I

When the optimization is

$$\operatorname{argmax}_{\Theta} \mathcal{P}(\mathbf{X}|\Theta)$$

we consider instead

$$\operatorname{argmax}_{\Theta} Q(\Theta|\Theta^{(t)})$$

where

$$Q(\Theta|\Theta^{(t)}) = -\sum_{\Omega_y} \mathcal{P}(Y = \mathbf{y}|\mathbf{X}, \Theta^{(t)}) \ln \mathcal{P}(\mathbf{X}|Y = \mathbf{y}, \Theta, \Theta^{(t)})$$

It is the expected value of the log likelihood function of the parameters Θ , with respect to the current conditional distribution of Y given X and the current estimates of the parameters $\Theta^{(t)}$.

- Θ : set of parameters
- I : random process modeling the observations (i.e. intensities)
- I : observation values (i.e. pixel intensities)
- Y : random variable modeling the hidden states (i.e. labels).
- Y : hidden state values (i.e. labels).
- T : matrix describing the probabilities of each values for each hidden

Exemple of Gaussian Mixture Models (GMM)

Exercise 41

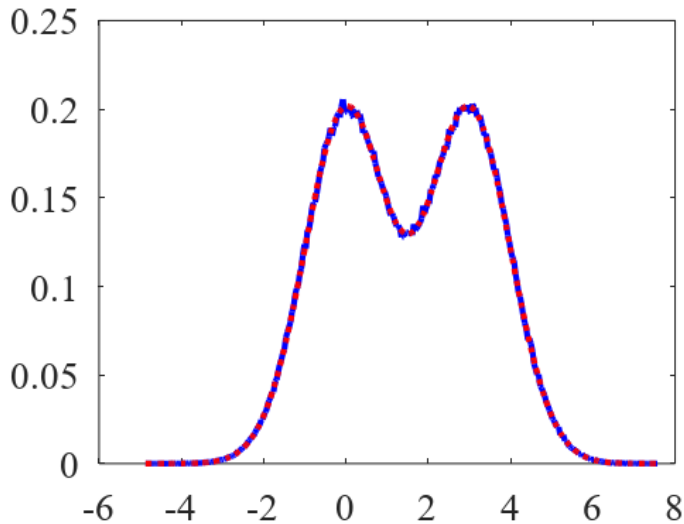
We consider the following pseudo-code, what is the probability distribution that is being sampled.

Require: N

Ensure: $l_0 \dots l_{N-1}$

```
1: for  $n=0:N-1$  do  
2:   Draw  $k$  a binary integer  
3:   Draw  $x$  a random value using  $\mathcal{N}(0,1)$   
4:   if  $k == 0$  then  
5:      $l_n = x$   
6:      $l_n = x + 3$ 
```

Answer to exercise 41 I



① We use the following notations

- K : binary random variable $\mathcal{P}(K = 0) = \mathcal{P}(K = 1) = 0.5$.
- X : Mixture of Gaussian random variable

$$\mathcal{P}(X = x|K = 0) = g_{0,1}(x) \text{ and } \mathcal{P}(X = x|K = 1) = g_{3,1}(x)$$

$$\begin{aligned}\mathcal{P}(X = x) &= \mathcal{P}(X = x|K = 0)\mathcal{P}(K = 0) + \mathcal{P}(X = x|K = 1)\mathcal{P}(K = 1) \\ &= 0.5g_{0,1}(x) + 0.5g_{3,1}(x)\end{aligned}$$

Exercise 42

Write a pseudocode simulating $\mathcal{P}(\sigma)$ and $Q(\sigma|\sigma^{(t)})$

Answer to exercise 42 I

- 1 Based on exercise 41, the probability one observation x_n is

$$f_{x_n=x_n|\sigma}(x_n) = 0.5g_{0,\sigma}(x_n) + 0.5g_{3,\sigma}(x_n)$$

The probability of all observations is

$$f_{\mathbf{x}|\sigma}(\mathbf{x}) = \prod_{n=0}^{N-1} 0.5g_{0,\sigma}(x_n) + 0.5g_{3,\sigma}(x_n)$$

In order to make computations within the 16 or 32 bits,

$$\ln f_{\mathbf{x}|\sigma}(\mathbf{x}) = \sum_{n=0}^{N-1} \ln (0.5g_{0,\sigma}(x_n) + 0.5g_{3,\sigma}(x_n))$$

- 2 At iteration t , the prior is

$$\mathcal{P}(Y_n = 0) = \sum_{n=0}^{N-1} 1(y_n^{(t)} = 0)$$

the posterior is

$$f_{Y_n=0|x_n,\sigma^{(t)}}(x_n) = \frac{\mathcal{P}(Y_n = 0)g_{0,\sigma^{(t)}}(x_n)}{\mathcal{P}(Y_n = 0)g_{0,\sigma^{(t)}}(x_n) + \mathcal{P}(Y_n = 1)g_{3,\sigma^{(t)}}(x_n)}$$

Answer to exercise 42 II

The log-likelihood given the hidden y_n

$$\ln f_{\mathbf{X}|\mathbf{Y}_n, \sigma^{(t)}}(\mathbf{X}) = -\ln(\sqrt{2\pi}\sigma^{(t)}) - \begin{cases} \frac{x^2}{2\sigma^{(t)2}} & \text{if } y_n = 0 \\ \frac{(x-3)^2}{2\sigma^{(t)2}} & \text{if } y_n = 1 \end{cases}$$

The expectation of this log-likelihood using the posterior is

$$Q_n = f_{Y_n=0|\mathbf{X}_n, \sigma^{(t)}}(x_n) \ln f_{\mathbf{X}=x_n|Y_n=0, \sigma^{(t)}}(x_n) \\ + f_{Y_n=1|\mathbf{X}_n, \sigma^{(t)}} \ln f_{\mathbf{X}|Y_n=1, \sigma^{(t)}}(x_n)$$

The function to be maximized is

$$Q(\sigma|\sigma^{(t)}, Y^{(t)}) = \sum_{n=0}^{N-1} Q_n$$

The error rate w.r. to the best possible prediction is

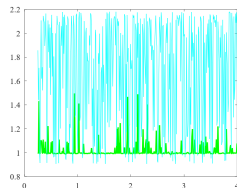
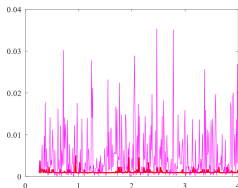
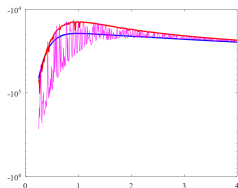
$$E = \frac{1}{N} \sum_{n=0}^{N-1} 1(y_n = 0 \Leftrightarrow x_n < 1.5)$$

Require: X

Ensure: σ

- 1: Choose randomly Y, σ ,
- 2: Store $Y^{\text{old}} := 0$
- 3: **while** $\exists y_n^{\text{old}} \neq y_n$ **do**
- 4: Store $Y^{\text{old}} := Y$
- 5: Compute $\mathcal{P}(Y_n = 0)$
- 6: Compute $\mathcal{P}(Y_n|x_n, \sigma^{(t)})$ with $\sigma^{(t)} = \sigma$
- 7: Find σ maximizing $Q(\sigma|\sigma^{(t)}, Y^{(t)})$ with $Y^{(t)} = Y$
- 8: Find $y_n = 1(\mathcal{P}(Y_n|x_n, \sigma^{(t)}) < 0.5)$
- 9: Compute $Q(\sigma|\sigma^{(t)}, Y^{(t)})$ with $\sigma^{(t)} = \sigma$ and $Y^{(t)} = Y$

Experimental results



In blue:

$$\ln \mathcal{P}(\mathbf{X} = X | \sigma)$$

In red:

$$Q(\sigma | \mathbf{X}, \sigma^{(t)}, y_n^{(t)})$$

$$\frac{1}{N} \sum_{n=0}^{N-1} \mathbf{1}(y_n = 0 \Leftrightarrow x_n < 1.5) \quad \sigma^{(t)}$$

- Horizontal axis indicates the value of σ .
- The thick lines in red and green are those obtained with many iterations.
- The thin lines in magenta and cyan are obtained at the first iteration.
- The initialization uses random values ($\sigma \mapsto \mathcal{U}(0.5, 4.5)$)

Third viewpoint on the Expectation-Minimization algorithm I

We need not use the correct probability distribution of the parameters. Instead of $\mathcal{P}(Y_n|\mathbf{X}, \Theta^{(t)})$, we may use any probability distribution $q(y_n)$ (or family of distributions).

- $q(y_n) = \mathcal{P}_Q(Y_n|x_n, \Theta^{(t)})$ is an inference model, its probability law is here denoted Q .
- $p(x_n, y_n) = \mathcal{P}(Y_n, X_n, \Theta^{(t)})$ is the joint distribution.
- $p'(y_n) = \mathcal{P}(Y_n|X_n, \Theta^{(t)})$ is the posterior distribution.
- The evidence lower bound (ELBO) is

$$\mathcal{E}_Q \left(\ln \frac{p(x_n, y_n)}{q(y_n)} \right)$$

- The Kullback-Leibler Divergence used here is

$$0 \leq \mathcal{D}_{\text{KL}}(q||p') = \mathcal{E}_Q \left(q(y_n) \ln \left(\frac{q(y_n)}{p'(y_n)} \right) \right)$$

Third viewpoint on the Expectation-Minimization algorithm II

Instead of maximizing

$$Q(\Theta|\Theta^{(t)}) = -\sum_{\Omega_y} \mathcal{P}(Y = \mathbf{y}|\mathbf{X}, \Theta^{(t)}) \ln \mathcal{P}(X|Y = \mathbf{y}, \Theta, \Theta^{(t)})$$

We maximize ELBO $-\mathcal{H}(q)$

$$-\sum_{\Omega_y} q(y_n) \ln \mathcal{P}(X|Y = \mathbf{y}, \Theta, \Theta^{(t)}) = \mathcal{E}_Q \left(\ln \frac{p(y_n)}{q(y_n)} \right) - \mathcal{H}_Q$$

because \mathcal{H}_Q does not depend on Θ and because

$$\text{ELBO} = \mathcal{P}(X_n) - \mathcal{D}_{\text{KL}} \leq \mathcal{P}(X_n)$$

This is a small proof:

$$\begin{aligned} \ln \mathcal{P}(X_n, \Theta) &= \mathcal{E}_Q \ln \mathcal{P}(X_n, \Theta) = \mathcal{E}_Q \ln \frac{\mathcal{P}(X_n, Y_n, \Theta)}{\mathcal{P}(Y_n|X_n, \Theta)} \\ &= \mathcal{E}_Q \ln \frac{\mathcal{P}(X_n, Y_n, \Theta)}{q(y_n)} + \mathcal{E}_Q \ln \frac{q(y_n)}{\mathcal{P}(Y_n|X_n, \Theta)} = \text{ELBO} + D_{\text{KL}} \end{aligned}$$

Exercise 43

We consider again the statistical model of exercise 39. We consider a discrete probability distribution depending on a parameter T for a given sample n , denoting here x_n and y_n as x and y .

$$q(y) = \begin{cases} 1(y = 0) & \text{if } x \leq T \\ 1(y = 1) & \text{if } x > T \end{cases}$$

- 1 Show that q is indeed a probability distribution whose entropy is zero.
- 2 Given the posterior of the probability distribution computed in exercise 40,

$$p'(y = 0) = \frac{\alpha g_{0,\sigma}(x)}{\alpha g_{0,\sigma}(x) + g_{3,\sigma}(x)} \text{ and } p'(y = 1) = 1 - p'(y = 0)$$

where σ denotes the estimated value $\sigma^{(t)}$. Compute the KL-divergence between the q and p' .

Using the E-M algorithm to make crisp choices II

Exercise

- 3 Show that

$$p'(y=0) > \frac{1}{2} \Leftrightarrow x < \frac{3}{2} + \frac{\sigma^2 \ln(\alpha)}{3}$$

- 4 Show that the KL-divergence is minimized when $T = \frac{3}{2} + \frac{\sigma^2 \ln(\alpha)}{3}$.

We now consider the whole dataset.

- 5 Compute $ELBO - \mathcal{H}_Q$
- 6 Compute $\sigma^{(t+1)}$ maximizing $ELBO - \mathcal{H}_Q$ as a function of $\mathcal{N}_0^{(t)}$ and $\mathcal{N}_1^{(t)}$ which are the set containing the samples $y_n^{(t)} = 0$ and $y_n^{(t)} = 1$. Show that

$$\sigma^{(t+1)} = \sqrt{\frac{1}{N} \left(\sum_{n \in \mathcal{N}_0^{(t)}} x_n^2 + \sum_{n \in \mathcal{N}_1^{(t)}} (x_n - 3)^2 \right)}$$

Answer to exercise 43 I

- 1 q is a discrete probability distribution with two possible values $y \in \{0, 1\}$ because

$$q(y) \geq 0 \text{ and } q(y=0) + q(y=1) = 1$$

The Entropy is

$$\mathcal{H}_Q = q(y=0) \ln \frac{1}{q(y=0)} + q(y=1) \ln \frac{1}{q(y=1)}$$

with the notation that $0 \times \ln 0 = 0$. $\mathcal{H}_Q = 0$ because first, $1 \times \ln 1 = 0$ and second, either $q(y=0) = 0$ and $q(y=1) = 1$ or $q(y=0) = 1$ and $q(y=1) = 0$.

- 2 The KL-divergence between q and p' is

$$\begin{aligned} \mathcal{D}_{\text{KL}}(q||p') &= q(y=0) \ln \frac{q(y=0)}{p'(y=0)} + q(y=1) \ln \frac{q(y=1)}{p'(y=1)} \\ &= 1(x \leq T) \ln \frac{1}{p'(y=0)} + 1(x > T) \ln \frac{1}{1-p'(y=0)} \\ &= -1(x \leq T) \ln \frac{\alpha g_{0,\sigma}(x)}{\alpha g_{0,\sigma}(x) + g_{3,\sigma}(x)} - 1(x > T) \ln \frac{g_{3,\sigma}(x)}{\alpha g_{0,\sigma}(x) + g_{3,\sigma}(x)} \end{aligned}$$

3

$$\begin{aligned}
 p'(y=0) > \frac{1}{2} &\Leftrightarrow \alpha g_{0,\sigma}(x) > \frac{\alpha}{2} g_{0,\sigma}(x) + \frac{1}{2} g_{3,\sigma}(x) \\
 &\Leftrightarrow \frac{\alpha}{2} g_{0,\sigma}(x) > \frac{1}{2} g_{3,\sigma}(x) \Leftrightarrow \frac{g_{0,\sigma}(x)}{g_{3,\sigma}(x)} > \frac{1}{\alpha} \\
 &\Leftrightarrow -\frac{x^2}{2\sigma^2} + \frac{(x-3)^2}{2\sigma^2} > -\ln(\alpha) \Leftrightarrow -6x + 9 > -2\sigma^2 \ln(\alpha) \\
 &\Leftrightarrow x < \frac{3}{2} + \frac{\sigma^2}{3} \ln(\alpha)
 \end{aligned}$$

4 We first prove that

$$\ln \frac{1}{p'(y=0)} < \ln \frac{1}{1-p'(y=0)} \Leftrightarrow x < \frac{3}{2} + \frac{\sigma^2}{3} \ln(\alpha)$$

The left statement is true iff

$$\begin{aligned}
 1 < \frac{\frac{1}{1-p'(y=0)}}{\frac{1}{p'(y=0)}} &= \frac{p'(y=0)}{1-p'(y=0)} \Leftrightarrow 1-p'(y=0) < p'(y=0) \\
 &\Leftrightarrow \frac{1}{2} < p'(y=0)
 \end{aligned}$$

Answer to exercise 43 III

Let us assume $x < \frac{3}{2} + \frac{\sigma^2}{3} \ln(\alpha)$, the former property proves that

$$\mathcal{D}_{\text{KL}}^{(\text{T} \leq x)}(q \| p') \geq \mathcal{D}_{\text{KL}}^{(\text{T} > x)}(q \| p')$$

Therefore $\text{T} \geq \frac{3}{2} + \frac{\sigma^2}{3} \ln(\alpha)$

Let us assume $x > \frac{3}{2} + \frac{\sigma^2}{3} \ln(\alpha)$, the former property proves that

$$\mathcal{D}_{\text{KL}}^{(\text{T} \leq x)}(q \| p') \leq \mathcal{D}_{\text{KL}}^{(\text{T} > x)}(q \| p')$$

Therefore $\text{T} \leq \frac{3}{2} + \frac{\sigma^2}{3} \ln(\alpha)$

This finally proves that the equality.

- 5 We set $\text{T} = \frac{3}{2} + \frac{\sigma^2}{3} \ln(\alpha)$, however, σ is actually $\sigma^{(t)}$. Let

$$\text{T}^{(t)} = \frac{3}{2} + \frac{(\sigma^{(t)})^2}{3} \ln(\alpha)$$

$$q(y_n = 0) = 1(x \leq \text{T}^{(t)}) \text{ and } q(y_n = 1) = 1(x > \text{T}^{(t)})$$

Whereas $f_{X_n = x_n | \sigma, y_n = 0}(x_n)$ depends on σ :

$$-\ln f_{X_n = x_n | \sigma, y_n = 0}(x_n) = -\ln g_{0, \sigma}(x_n) = \ln(\sqrt{2\pi}\sigma) + \frac{x_n^2}{2\sigma^2}$$

Answer to exercise 43 IV

And

$$-\ln f_{\tilde{x}_n=x_n|\sigma, y_n=1}(x_n) = -\ln g_{3,\sigma}(x_n) = \ln(\sqrt{2\pi}\sigma) + \frac{(x_n - 3)^2}{2\sigma^2}$$

So $J = \text{ELBO} - \mathcal{H}_Q$ is

$$\begin{aligned} J &= \sum_{n=0}^{N-1} (-1(x \leq \tau^{(t)}) \ln f_{\tilde{x}_n=x_n|\sigma, y_n=0}(x_n) \\ &\quad - 1(x > \tau^{(t)}) \ln f_{\tilde{x}_n=x_n|\sigma, y_n=1}(x_n)) \\ &= N \ln(\sqrt{2\pi}\sigma) + \sum_{n=0}^{N-1} 1(x_n \leq \tau^{(t)}) \frac{x_n^2}{\sigma^2} \\ &\quad + \sum_{n=0}^{N-1} 1(x_n > \tau^{(t)}) \frac{(x_n-3)^2}{\sigma^2} \end{aligned}$$

6 To find the value of σ minimizing J , we compute

$$\frac{\partial J}{\partial \sigma} = \frac{N}{\sigma} - \frac{2}{\sigma^3} \sum_{n=0}^{N-1} 1(x_n \leq \tau^{(t)}) \frac{x_n^2}{2} + 1(x_n > \tau^{(t)}) \frac{(x_n - 3)^2}{2}$$

Canceling this derivative yields a value of σ now called $\sigma^{(t+1)}$

$$\left(\sigma^{(t+1)}\right)^2 = \frac{1}{N} \sum_{n=0}^{N-1} \left(1(x_n \leq \tau^{(t)}) x_n^2 + 1(x_n > \tau^{(t)}) (x_n - 3)^2\right)$$

Note that because q is actually modeling a deterministic function
 $1(x_n \leq T^{(t)}) \Leftrightarrow y_n^{(t)} = 0 \Leftrightarrow n \in \mathcal{N}_0^{(t)}$

- Entropy: $\mathcal{H}(p) = -\sum_{x_i} p(x_i) \ln p(x_i) \geq 0$ There exists also a differential definition of \mathcal{H} .
- KL-divergence: $\mathcal{D}_{KL}(q||p) = \sum_{x_i} q(x_i) \ln \frac{q(x_i)}{p(x_i)} \geq 0$ There exists also a differential definition of \mathcal{D}_{KL} .
- ELBO = $\ln \left(\frac{p(x,y)}{q(y)} \right)$

Conclusion of subsection 3, Probabilities as a loss function designer

- Difference between \mathcal{P} and f
- Expectation-maximization = soft assignment with a probabilistic interpretation
- Simplifies only the local probabilistic expression
- First viewpoint: Concavity based lower-bound
- Second viewpoint: Average value of the log-likelihood with weights equal to the probability of a hidden parameters given previous iteration.
- Third viewpoint: Average value of the log-likelihood with weights equal to probabilities chosen so as to minimize a distance with the probability of a hidden parameter given previous iteration.

Table of Contents I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

Table of Contents II

8. Supplementary material regarding matrices

Content of section 6, Curse of dimensionality, regularization and sparsity I

6.1 Data preparation

6.2 Feature construction

6.3 Kernel trick

6.4 Curse of dimensionality and feature extraction

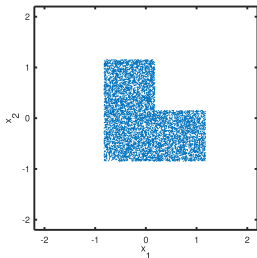
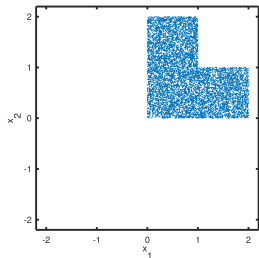
6.5 Principal Component Analysis

6.6 Supervised feature extraction

6.7 Regularization

6.8 Feature selection

Centering the feature matrix



A centered feature matrix fulfills

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

Exercise 44

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 44

Let β_f be defined as

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

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

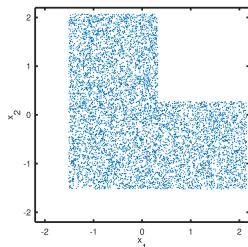
$$\sum_{n=1}^N X'_{nf} = \sum_{n=1}^N (X_{nf} - \beta_f) = \sum_{n=1}^N X_{nf} - \beta_f = 0$$

Normalizing features

Normalizing means

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

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



Exercise 45

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

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

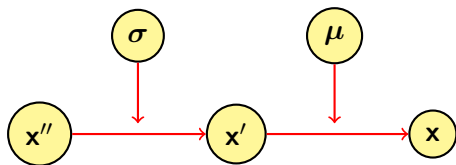
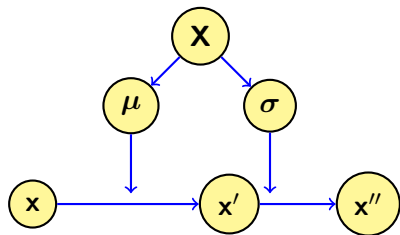
where $x'_{nf} = \alpha_f x_{nf}$

Answer to exercise 45

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

we get

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



Exercise 46

The exercises 44 and 45 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 \mathbf{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 $\mathbf{HX} (\text{diag}(\mathbf{X}^T \mathbf{H}^2 \mathbf{X}))^{-\frac{1}{2}}$ is centered and normalized.
- 3 Write the Matlab/Octave implementation of $\mathbf{HX} (\text{diag}(\mathbf{X}^T \mathbf{H}^2 \mathbf{X}))^{-\frac{1}{2}}$
 $(\text{diag}(A))_{ij} = a_{ij} 1(j = i)$ and $((\text{diag}(A))_{ij})^{-\frac{1}{2}} = \frac{1}{\sqrt{a_{ij}}} 1(j = i)$

Matrix formulas

- Column number j of a matrix \mathbf{A} :

$$\begin{bmatrix} a_{1j} \\ \vdots \\ a_{lj} \end{bmatrix}$$

- Row number i of a matrix \mathbf{A} :

$$[a_{i1}, \dots, a_{ij}]$$

- Left-multiplication of \mathbf{A} by a diagonal matrix $\mathbf{D} = [d_i 1(j=i)]_{ij}$:

$$(\mathbf{DA})_{ij} = d_i a_{ij}$$

- Right-multiplication of \mathbf{A} by a diagonal matrix $\mathbf{D} = [d_i 1(j=i)]_{ij}$:

$$(\mathbf{AD})_{ij} = a_{ij} d_j$$

- Multiplication of two matrices

$$(\mathbf{AB})_{ij} = \sum_k a_{ik} b_{kj}$$

- Left-multiplication of \mathbf{B} by \mathbf{A}^T

$$(\mathbf{A}^T \mathbf{B})_{ij} = \sum_k a_{ki} b_{kj}$$

Answer to exercise 46 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 46 II

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

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

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

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

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

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 $\mathbf{H} = \text{eye}(N) - 1/N * \text{ones}(N)$;

$$\mathbf{X}_p = \mathbf{H} * \mathbf{X} * \text{diag}(\text{diag}(\mathbf{X}' * \mathbf{H} * \mathbf{H} * \mathbf{X}) . ^{-1/2}) ;$$

- $\text{diag}(\mathbf{A})$ is a diagonal matrix composed of the diagonal components of \mathbf{A} .
- $\text{diag}(\mathbf{A})^\alpha$ when the diagonal components are positive is equal to A_{ii}^α .

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 6, Curse of dimensionality, regularization and sparsity I

6.1 Data preparation

6.2 Feature construction

6.3 Kernel trick

6.4 Curse of dimensionality and feature extraction

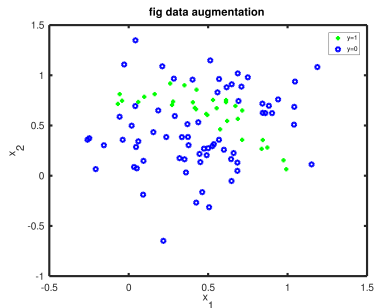
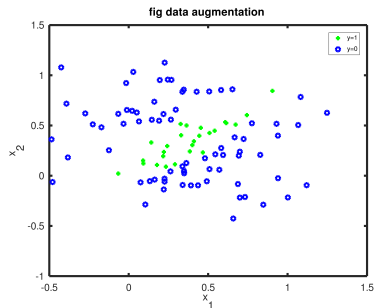
6.5 Principal Component Analysis

6.6 Supervised feature extraction

6.7 Regularization

6.8 Feature selection

Can we classify the following datasets with a linear classifier?



Yes

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

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

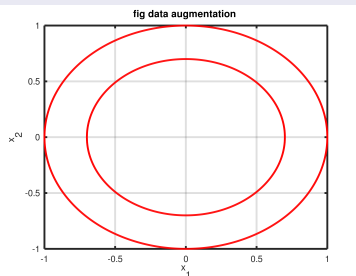
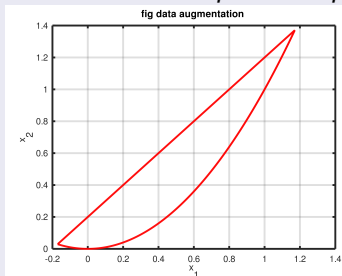
Notations

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

Examples of linear classifiers I

Exercise 47

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.

Exercise

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

$$f(\vec{\mathbf{x}}) = 1(b_1 - \mathbf{a}_1 \cdot \vec{\mathbf{x}})1(b_2 - \mathbf{a}_2 \cdot \vec{\mathbf{x}})$$

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

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

$$f(\vec{\mathbf{x}}) = 1(b_1 - \mathbf{a}_1 \cdot \vec{\mathbf{x}})1(b_2 - \mathbf{a}_2 \cdot \vec{\mathbf{x}})$$

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

Answer to exercise 47 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.

$$x_3 = x_1^2$$

$$x_4 = x_1 x_2$$

$$x_5 = x_2^2$$

- ④ The delimiting equations can be written as

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

$$0 + \bar{x}_2 - \bar{x}_3 \geq 0$$

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

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

The delimiting equations can be written as

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

$$1 - \bar{x}_3 - \bar{x}_5 \geq 0$$

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

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

$$\overset{\omega}{\mathcal{F}} \neq \omega(\mathcal{F})$$

We introduce some new notations

- $\omega(\mathbf{x})$ is the constructed feature vector.
- ω is a mapping of \mathcal{F} into $\overset{\omega}{\mathcal{F}}$
(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 $\forall \overset{\omega}{\mathbf{x}}, \exists \mathbf{x}, \overset{\omega}{\mathbf{x}} = \omega(\mathbf{x})$.
- $\| \cdot \|$ is the Euclidean norm of \mathcal{F} and $\| \cdot \|_{\omega}$ is the Euclidean norm of $\overset{\omega}{\mathcal{F}}$.

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

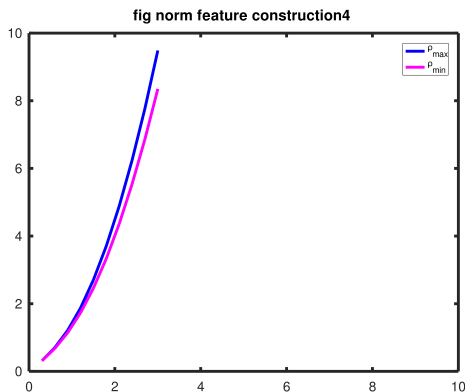
The samples in the dataset is inside $\omega(\mathcal{F})$. However they are considered as members of the 5D-space denoted $\overset{\omega}{\mathcal{F}}$.

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.

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

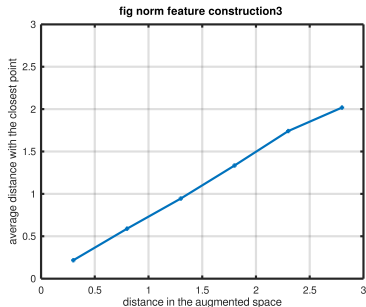


Most points in \mathcal{F} are far from $\omega(\mathcal{F})$

Average distance between points in \mathcal{F} and points that can be mapped from \mathcal{F} with ω .

$$d(t) = \mathcal{E} \left[\min_{\mathbf{x}' \in \mathcal{F}} \left\{ \|\omega(\mathbf{x}') - \bar{\mathbf{x}}\|_{\omega} \mid \|\bar{\mathbf{x}}\|_{\omega} = t \right\} \right]$$

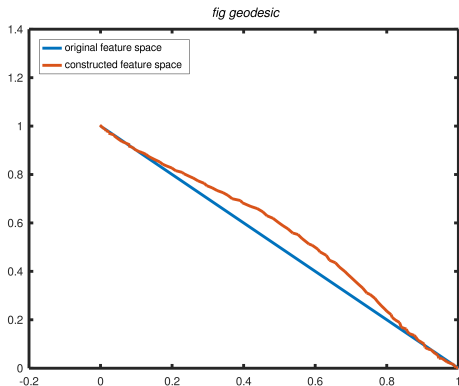
where \mathcal{E} is expected value when following here the uniform law.



The closest point are close to where we expect them

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

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



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

New notations

- \mathcal{F} and $\overset{\omega}{\mathcal{F}}$
- \mathbf{x} and $\overset{\omega}{\mathbf{x}}$
- $\| \cdot \|$ and $\| \cdot \|_{\omega}$
- ω

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
 - $\| \cdot \|_{\omega}$ is different in nature from $\| \cdot \|$
 - $\| \cdot \|$ is different in value from $\| \omega(\cdot) \|_{\omega}$
 - 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 6, Curse of dimensionality, regularization and sparsity I

- 6.1 Data preparation
- 6.2 Feature construction
- 6.3 Kernel trick**
- 6.4 Curse of dimensionality and feature extraction
- 6.5 Principal Component Analysis
- 6.6 Supervised feature extraction
- 6.7 Regularization
- 6.8 Feature selection

Exercise 48

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 K

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

- 1 Express K as function of $[x_1, x_2]$ and $[x'_1, x'_2]$. Is it left-linear, right-linear?
- 2 Compute $\mathbf{K} = [K(\mathbf{x}_m, \mathbf{x}_n)]_{m,n}$
- 3 Show that the inverse of \mathbf{K} is defined?

Exercise

The inverse of \mathbf{K} is

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

We define

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

- 4 Compute $K(\mathbf{x}_1)$, $K(\mathbf{x}_2)$ and $K(\mathbf{x}_3)$.
- 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 $K(\mathbf{x}_1 - \mathbf{x}_2)$.

Answer to exercise 48 I

1

$$K(\mathbf{x}, \mathbf{x}') = x_1 x'_1 + x_2 x'_2 + x_1^2 x'_1{}^2 + x_1 x'_1 x_2 x'_2 + x_2^2 x'_2{}^2$$

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

$$(K)_{11} = K([1 \ 0], [1 \ 0]) = 1 \times 1 + 0 + 1^2 \times 1^2 + 0 + 0$$

$$(K)_{12} = K([1 \ 0], [0 \ 1]) = 1 \times 0 + 0 \times 1 + 1^2 \times 0^2 + 1 \times 0 \times 0 \times 1 + 0^2 \times 1^2$$

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

3 K is invertible because $\det(K) \neq 0$.

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

Answer to exercise 48 II

- 4 $[K(\mathbf{x}_1, \mathbf{x}_m)]_m$ is the first line of \mathbf{K} , $[2, 0, 2]$ so
 $K(\mathbf{x}_1) = [K(\mathbf{x}_1, \mathbf{x}_m)]_m \mathbf{K}^{-1} = [1, 0, 0]$ and $K(\mathbf{x}_1)\omega(\mathbf{X}) = [1, 0, 1, 0, 0]$
 $[K(\mathbf{x}_2, \mathbf{x}_m)]_m$ is the first line of \mathbf{K} , $[0, 2, 2]$ so
 $K(\mathbf{x}_2) = [K(\mathbf{x}_2, \mathbf{x}_m)]_m \mathbf{K}^{-1} = [0, 1, 0]$ and $K(\mathbf{x}_2)\omega(\mathbf{X}) = [0, 1, 0, 0, 1]$
 $[K(\mathbf{x}_3, \mathbf{x}_m)]_m$ is the first line of \mathbf{K} , $[2, 2, 5]$ so
 $K(\mathbf{x}_3) = [K(\mathbf{x}_3, \mathbf{x}_m)]_m \mathbf{K}^{-1} = [0, 0, 1]$ and $K(\mathbf{x}_3)\omega(\mathbf{X}) = [1, 1, 1, 1, 1]$
- 5 Let us consider $\mathbf{x}' = [1, -1]$.
 $\omega(\mathbf{x}') = [1, -1, 1, -1, 1]$
- 6 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.

7

$K(\mathbf{x}') = [\omega(\mathbf{x}') \cdot \omega(\mathbf{X})]_n \mathbf{K}^{-1} = [2, 0, -1] \mathbf{K}^{-1} = [2, 1, -1]$
and $K(\mathbf{x}')\omega(\mathbf{X}) = [1, 0, 1, -1, 0]$ Now we want to show that

$$K(\mathbf{x}')\omega(\mathbf{X}) \notin \omega(\mathcal{F})$$

If this was wrong then there would exist 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.

Basic idea

$\mathbf{x} \cdot \mathbf{x}'$ is replaced by $K(\mathbf{x}, \mathbf{x}')$

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

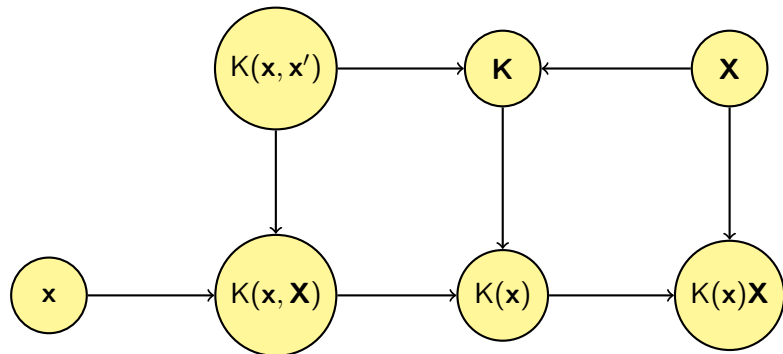
Samples act as a basis

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



- Kernel matrix

$$\mathbf{K} = [\mathbf{K}(\mathbf{x}_m, \mathbf{x}_n)]_{nm}$$

- Kernel values on the dataset as a row vector

$$[\mathbf{K}(\mathbf{x}, (\mathbf{X})_n)]_n$$

- Mapping in the kernel-induced space

$$\mathbf{K}(\mathbf{x})$$

- Back to the feature space

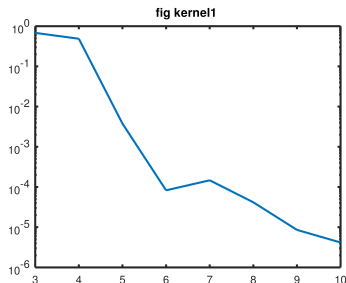
$$\mathbf{K}(\mathbf{x})\mathbf{X}$$

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



Exercise 49

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

Answer to exercise 49 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 $\mathbf{x}' = [\omega(\mathbf{x}), \omega(\mathbf{X}(1, :)) \dots] \mathbf{K}^{-1}$
- 10: Update d with $d := d + \|\mathbf{x}' \omega(\mathbf{X}) - \omega(\mathbf{x})\|_{\mathcal{F}}$.
- 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 6, Curse of dimensionality, regularization and sparsity I

- 6.1 Data preparation
- 6.2 Feature construction
- 6.3 Kernel trick
- 6.4 Curse of dimensionality and feature extraction**
- 6.5 Principal Component Analysis
- 6.6 Supervised feature extraction
- 6.7 Regularization
- 6.8 Feature selection

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 = 1.56642  
warning: called from
```

Example of this phenomenon

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

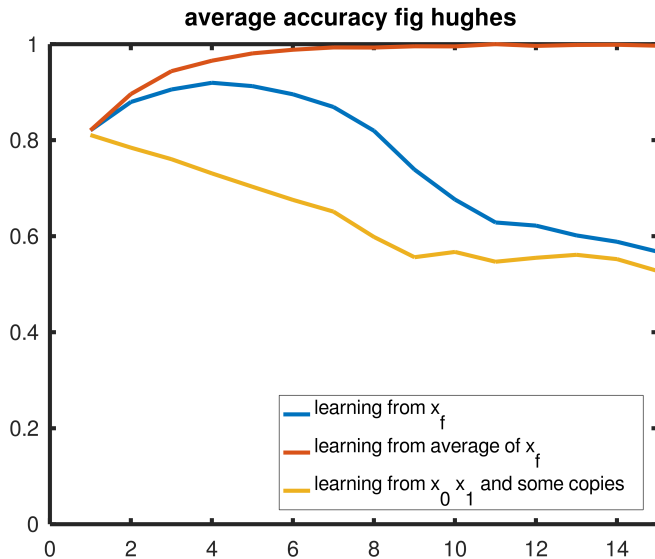
$$\hat{y} \sim \mathcal{U}(\{0, 1\}) \text{ and } \hat{\mathbf{x}}|_{y=0} \sim \mathcal{N}(-1, 1) \quad \hat{\mathbf{x}}|_{y=1} \sim \mathcal{N}(1, 1)$$

Require: F dimension of feature space

Ensure: A_1, A_2, 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 A_1 with Y_t and \mathbf{w}_1 -predictions.
- 7: Compute 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 A_3 with Y_t and \mathbf{w}_2 -predictions.
- 12: Average A_1, A_2, A_3 .

Simulations



- Experiment

With feature extraction, we try to find linear combinations of existing features that captures most information.

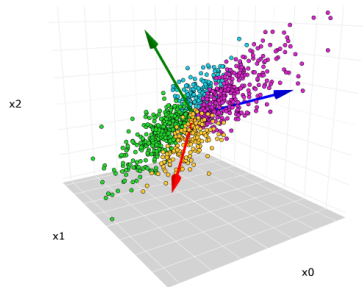
- Experiment

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.

- 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.

- \mathbf{x} are drawn with respect to y
- The obtained \mathbf{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 6, Curse of dimensionality, regularization and sparsity I

- 6.1 Data preparation
- 6.2 Feature construction
- 6.3 Kernel trick
- 6.4 Curse of dimensionality and feature extraction
- 6.5 Principal Component Analysis**
- 6.6 Supervised feature extraction
- 6.7 Regularization
- 6.8 Feature selection

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

$$\mathbf{x} \rightarrow P, D$$

Matrix computations

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

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

$$D, F_1 \rightarrow A_{\text{TPCA}}$$

Analysis and synthesis

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

Approximation

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

Accuracy of approximation

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

What we get with PCA I

Analysis

Given \mathbf{x} , we transform into

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

Components are statistically independent from each others

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

We can construct approximations by truncating the vector.

$$\text{PCA}_{\mathcal{T}\mathcal{P}}(\mathbf{x}) = \mathbf{x}\mathbf{P}\text{diag}(\overset{\langle -F_1 - \rangle}{[1 \dots 1, 0 \dots 0]})$$

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 $\text{PCA}_{\mathcal{T}\mathcal{P}}(\mathbf{x})$, we get a good approximation of \mathbf{x} , denoted $\text{PCA}_{\mathcal{T}}(\mathbf{x})$

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

We also have an orthogonality property

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

The accuracy of the approximation is

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

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

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

$$A_{\text{TPCA}} = \frac{\sum_{f=1}^{F_1} \lambda_f}{\sum_{f=1}^F \lambda_f} = \frac{1}{\text{tr}(D)} \sum_{f=1}^{F_1} \lambda_f$$

Illustrating the notations in a toy example I

Exercise 50

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 $\|\mathbf{x}\|^2$, $\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2$*
- 6 *Compute $A_{\mathcal{T}PCA}$*
- 7 *Check the **X**-signification of $A_{\mathcal{T}PCA}$*

Answer to exercise 50 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

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

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

2

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

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

$$\text{PCA}_{\mathcal{P}}(\mathbf{x}) = \mathbf{xP} = [\mathbf{x}\mathbf{e}_1^T \quad \mathbf{x}\mathbf{e}_2^T] = \begin{bmatrix} \frac{\sqrt{2}}{2}(x_1 + x_2) & \frac{\sqrt{2}}{2}(x_1 - x_2) \end{bmatrix}$$

Denoting the component of $\text{PCA}_{\mathcal{P}}(\mathbf{x})$ as x'_1, x'_2 , we get for the synthesis

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

Answer to exercise 50 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

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

We have shown previously

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

As we keep only the first component,

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

Answer to exercise 50 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

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

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 50 V

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

$$A_{\mathcal{T}PCA} = \frac{1}{1 + \frac{1}{9}} = \frac{9}{10}$$

- 7 The signification of $A_{\mathcal{T}PCA}$ 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

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

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

A F -multivariate distribution is defined with a mean $\boldsymbol{\mu}$ and a **covariance matrix** $\boldsymbol{\Sigma}$

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

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

Notation

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

Exercise 51

We consider two independent 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

$$\overset{r}{\mathbf{x}} = \begin{bmatrix} \frac{2}{3}\overset{r}{z}_1 + \frac{1}{3}\overset{r}{z}_2, & \frac{1}{3}\overset{r}{z}_1 + \frac{2}{3}\overset{r}{z}_2 \end{bmatrix}$$

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

Answer to exercise 51 I

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{2}{3}\dot{z}_1 + \frac{1}{3}\dot{z}_2, & \frac{1}{3}\dot{z}_1 + \frac{2}{3}\dot{z}_2 \end{bmatrix}$$

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

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

Because these are independent Gaussian distributions, we have

$$\mathcal{E}[(\dot{z}_1)^2] = \mathcal{E}[(\dot{z}_2)^2] = 1 \text{ and } \mathcal{E}[\dot{z}_1\dot{z}_2] = 0$$

So we get

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

Exercise 52

We consider a centered multivariate normal distribution

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

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

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

Exercise

- 4 We now use polar coordinates $x_1 = r \cos(\theta)$ and $x_2 = r \sin(\theta)$. Show that

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

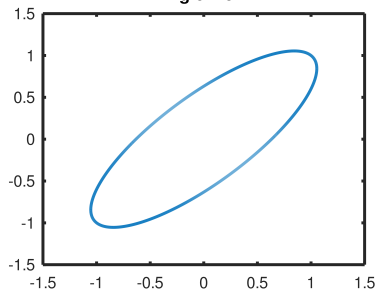
and hence that a parametric description of the contour is

$$\begin{cases} x(\theta) = r(\theta) \cos(\theta) \\ y(\theta) = r(\theta) \sin(\theta) \end{cases}$$

- 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.

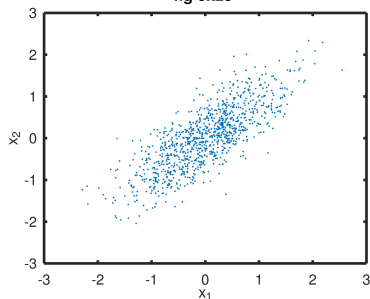
Using the theoretical equations,

fig ex25



By drawing 1000 points of \hat{z}_1^r , \hat{z}_2^r ,
and computing \mathbf{x} ,

fig ex25



Answer to exercise 52 I

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

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

$$f_{\mathbf{x}}(\mathbf{x}) = \frac{1}{2\pi|\det(\Sigma)|} e^{-J}$$
$$2J = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$
$$2J = 5x_1x_1 + -4x_1x_2 + -4x_2x_1 + 5x_2x_2$$
$$2J = 5x_1^2 - 8x_1x_2 + 5x_2^2$$

- 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 52 II

3

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

4

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

we get

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

And finally

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

Answer to exercise 52 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 $\mathbf{x} \sim \mathcal{N}(0, \Sigma)$

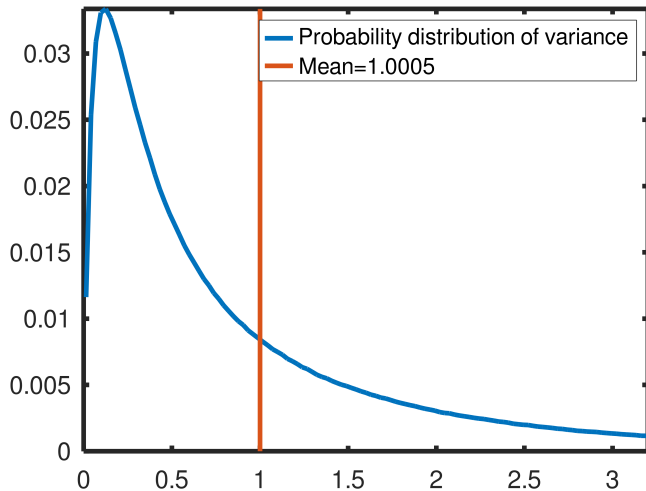
$$\text{var}(\hat{\mathbf{x}}) = \mathcal{E} \left[\hat{\mathbf{x}}(\hat{\mathbf{x}})^T \right] = \text{tr}(\Sigma)$$

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 $\text{tr}(\Sigma)$.

fig explain variance



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

$$\mathcal{E}[\mathbf{x}'(\mathbf{x}')^T] = \text{var}(\mathbf{x}') + \mathcal{E}[\mathbf{x}']\mathcal{E}[\mathbf{x}']^T = \text{tr}(\Sigma) + \boldsymbol{\mu}\boldsymbol{\mu}^T$$

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

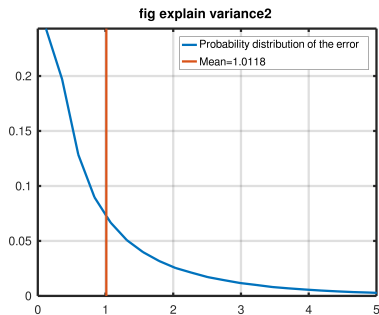
$$\mathcal{E} \left[\|\hat{\mathbf{x}} - \text{PCA}_{\mathcal{T}}(\hat{\mathbf{x}})\|^2 \right] = (1 - A_{\mathcal{T}, \text{PCA}}(\Sigma, F_1)) \text{tr}(\Sigma)$$

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

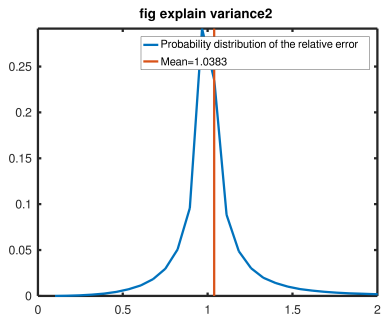
An experiment

- 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 $A(i) := A_{\mathcal{T}, \text{PCA}}$
- 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 $b(i) := (\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2) / \|\mathbf{x}\|^2$
- 8: Plot histogram of $\frac{a(i)}{1-A(i)}$ and of $\frac{b(i)}{1-A(i)}$

Variance 2

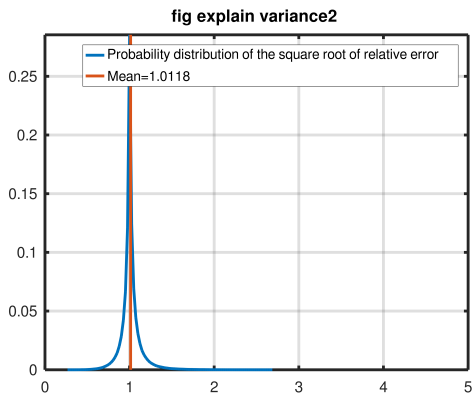


$$\frac{\|\mathbf{x} - \text{PCA}_{\mathcal{T}}(\mathbf{x})\|^2}{1 - A_{\mathcal{T}, \text{PCA}}}$$



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

Probability distribution of the square root of the relative error



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

Frobenius norm and variance

- $\|\cdot\|_{\mathcal{F}}^2$ has a definition using trace.

$$\|\mathbf{X}\|^2 = \text{tr}(\mathbf{X}^T \mathbf{X})$$

- $\|\cdot\|_{\mathcal{F}}$ 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 $\mathbf{X}^T \mathbf{X}$

$$\|\mathbf{X}\|_{\mathcal{F}}^2 = \text{tr}(D) = \sum_{f=1}^F \lambda_f$$

- 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 $[\mathbf{x}_n \mathbf{x}_{n'}^T]_{n, n'}$.

$$\|\mathbf{X}\|_F^2 = \text{tr}(\mathbf{X}^T \mathbf{X}) = \text{tr}(\mathbf{X} \mathbf{X}^T)$$

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.

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

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 52 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 359.

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 53

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 53 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 $\mathbf{x} = [1 \quad -1]$,

$$\mathbf{x}\Sigma = \left[\frac{1}{9} \quad \frac{1}{9}\right] = \mathbf{x}$$

So $\mathbf{e}_2 = [1 \quad -1] \frac{\sqrt{2}}{2}$ is the second eigenvector.

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

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

Statistic

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

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

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.

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

with

$$\mathbf{X}^T \mathbf{X} = \mathbf{PDP}^T$$

We get independent normalized
Gaussian random variables

$$z_f \sim \mathcal{N}(0, 1)$$

and a white covariance matrix

$$\mathbf{Z}^T \mathbf{Z} = \mathbf{1}$$

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

We get unitary random components

$$\forall f, \quad \text{var}(x'_f) = 1$$

And unitary column vectors

$$\|\mathbf{X}'_f\| = 1$$

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 54

We consider the same centered multivariate normal distribution as defined in exercise 52.

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

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

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

- 1 Write the equations of the whitening process transforming \mathbf{x}^r into \mathbf{z}^r .

We now assume as in exercise 51 that actually \mathbf{x}^r comes from two centered normalized Gaussian random variable z_1^r and z_2^r .

$$x_1^r = \frac{2}{3}z_1^r + \frac{1}{3}z_2^r \text{ and } x_2^r = \frac{1}{3}z_1^r + \frac{2}{3}z_2^r$$

- 2 Check that \mathbf{z}^r is indeed white.

Answer to exercise 54 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 $\mathbf{z}' = \mathbf{xPD}^{-1/2}$

$$z'_1 = \frac{\sqrt{2}}{2}(x_1 + x_2)$$

$$z'_2 = \frac{3\sqrt{2}}{2}(x_1 - x_2)$$

- 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.

$$\text{corr}\mathbf{X} = \text{cov norm}\mathbf{X}$$

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

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

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.

Exercise 55

We consider the tiny dataset of exercise 50 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 the correlation matrix.*

Answer to exercise 55 I

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

1

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

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

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 6, Curse of dimensionality, regularization and sparsity I

- 6.1 Data preparation
- 6.2 Feature construction
- 6.3 Kernel trick
- 6.4 Curse of dimensionality and feature extraction
- 6.5 Principal Component Analysis
- 6.6 Supervised feature extraction**
- 6.7 Regularization
- 6.8 Feature selection

Transforming PCA into a supervised feature extraction technique

To have zero mean, we consider \tilde{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

$$\text{cov}(\overset{r}{\mathbf{x}}\overset{r}{\tilde{y}}) = [\mathcal{E}(\overset{r}{x_1}\overset{r}{\tilde{y}}) \dots \mathcal{E}(\overset{r}{x_F}\overset{r}{\tilde{y}})]$$

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

PCA with a modification on the covariance matrix

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

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

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

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

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

The new PCA supervised-methodology

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

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 $\text{diag}(\tilde{Y})\mathbf{X}$ instead of \mathbf{X} .
- 3 We have also seen in frame 221 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 6, Curse of dimensionality, regularization and sparsity I

- 6.1 Data preparation
- 6.2 Feature construction
- 6.3 Kernel trick
- 6.4 Curse of dimensionality and feature extraction
- 6.5 Principal Component Analysis
- 6.6 Supervised feature extraction
- 6.7 Regularization**
- 6.8 Feature selection

What it is

About the previous examples of regularization

We had to inverse an ill-conditioned matrix and to achieve this we add λI 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 decomposition.

$$A = UDV'$$

Exercise 56

- 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 along features
→

↓ smoothing along samples

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

Answer to exercise 56 I

Random vectors stacked in \mathbf{X} .

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

When drawn, the condition number is okay because,

$$(\mathbf{X}^T \mathbf{X})_{mn} \approx N1(m = n)$$

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 56 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

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

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 bandwidths.
- However using test samples very close to training samples is an issue.

Modified loss function

L_2 -regularization consists in adding

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

with $\lambda > 0$ a cost parameter.

This is called the **ridge** OLS.

OLS stands for **Ordinary Least Square**.

Exercise 57

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

Answer to exercise 57 I

$$\begin{aligned}2\mathcal{L}_r(\mathcal{S}, f^v) &= \left(\hat{\mathbf{X}}\mathbf{w}^T\right)^T \left(\hat{\mathbf{X}}\mathbf{w}^T\right) - \left(\hat{\mathbf{X}}\mathbf{w}^T\right)^T \tilde{\mathbf{Y}} \\ &\quad - \tilde{\mathbf{Y}}^T \left(\hat{\mathbf{X}}\mathbf{w}^T\right) + \tilde{\mathbf{Y}}^T \tilde{\mathbf{Y}} + \lambda \mathbf{w}\mathbf{w}^T \\ 2\mathcal{L}(\mathcal{S}, f^v) &= \mathbf{w} \left(\hat{\mathbf{X}}^T \hat{\mathbf{X}} + \lambda \mathbf{I}\right) \mathbf{w}^T - 2\mathbf{w} \hat{\mathbf{X}}^T \tilde{\mathbf{Y}} + \tilde{\mathbf{Y}}^T \tilde{\mathbf{Y}}\end{aligned}$$

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

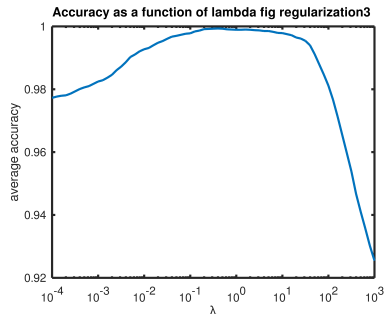
$$\mathbf{w}^T = \left(\hat{\mathbf{X}}^T \hat{\mathbf{X}} + \lambda \mathbf{I}\right)^{-1} \hat{\mathbf{X}}^T \tilde{\mathbf{Y}}$$

An experiment showing increased performance with L_2 -regularization

Require: λ

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

- 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



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

Exercise 58

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.

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

- 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 \hat{a} maximizing the posterior probability is defined as

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

- ① Denoting the likelihood of Y given X and a

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

$\sigma_\eta^2 \mathbf{I}_N$ 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.

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

Answer to exercise 58 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

$$f_{a|Y,X}^r(X, Y, a) = \frac{f_{Y|a,X}^r(X, Y, a)f_a^r(a)}{\int_{-\infty}^{+\infty} f_{Y|a,X}^r(X, Y, a)f_a^r(a) da}$$

- 3 Because the denominator depends only of X and Y , it is possible to denote its logarithm $-Z(X, Y)$ and hence to get

$$\ln f_{a|Y,X}^r(X, Y, a) = Z(X, Y) + \ln f_{Y|a,X}^r(X, Y, a) + \ln f_a^r(a)$$

There exists a quantity κ not depending on X and Y such that

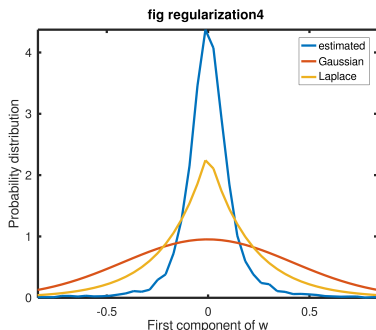
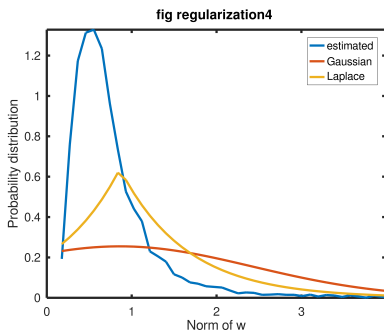
$$\begin{aligned} \ln f_{a|Y,X}^r(X, Y, a) &= Z(X, Y) + \kappa - \frac{1}{2\sigma_\eta^2} (Y - aX)^T (Y - aX) - \frac{1}{2\sigma_a^2} a^2 \\ &= Z(X, Y) + \kappa - \frac{1}{\sigma_\eta^2} \left((Y - aX)^T (Y - aX) + \frac{\sigma_\eta^2}{\sigma_a^2} a^2 \right) \end{aligned}$$

Probability distribution of the learned parameters

Prior modeling Choice of the prior

$$\mathbf{w}^T = \begin{pmatrix} \Delta^T & \Delta \\ \mathbf{X} & \mathbf{X} \end{pmatrix}^{-1} \Delta^T Y$$

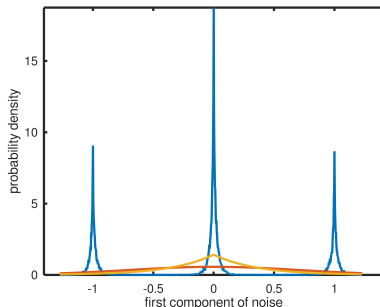
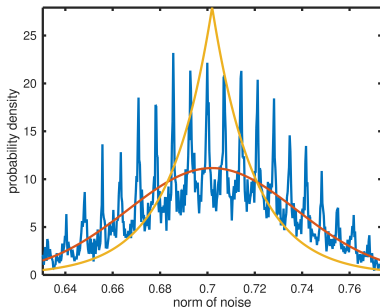
$$\Rightarrow \mathbf{w}_f \sim \mathcal{L}(0, 0.2) \text{ or } \mathbf{w}_f \sim \mathcal{N}(0, 0.4)$$



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

$$\eta = Y - \mathbf{X}\mathbf{w} \quad \text{with} \quad \mathbf{w}^T = \begin{pmatrix} \Delta^T & \Delta^T \\ \mathbf{X} & \mathbf{X} \end{pmatrix}^{-1} \Delta^T Y$$

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



Two kinds of regularization for OLS

LASSO

Least absolute shrinkage and selection operator

It is a Laplacian approximation of the parameter prior.

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

Ridge OLS

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

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

Laplace function

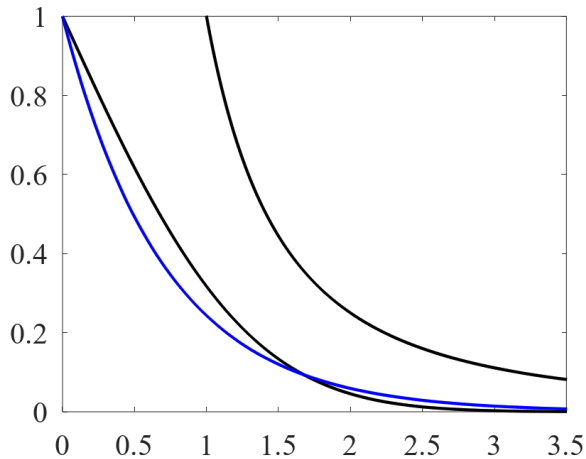


Figure 9:

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 6, Curse of dimensionality, regularization and sparsity I

- 6.1 Data preparation
- 6.2 Feature construction
- 6.3 Kernel trick
- 6.4 Curse of dimensionality and feature extraction
- 6.5 Principal Component Analysis
- 6.6 Supervised feature extraction
- 6.7 Regularization
- 6.8 Feature selection**

Exercise 59

We consider again exercise 25 and the proposed solution in exercise 27 where

$$\hat{\Delta} = [\mathbf{X}1], \quad \mathbf{w} = [-\mathbf{a} \ b] \text{ and } \mathbf{w}^T = \left(\begin{array}{c|c} \Delta^T & \Delta \\ \hline \mathbf{X} & \mathbf{X} \end{array} \right)^{-1} \begin{array}{c} \Delta^T \\ \mathbf{X}^T \end{array} \tilde{\mathbf{Y}}$$

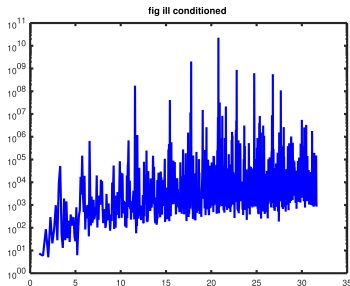
with

$$f_{\mathbf{a},b}(\mathbf{x}) = 1(\mathbf{a} \cdot \mathbf{x} \leq b)$$

- 1 Let us suppose that the first component of all samples in \mathcal{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 59 I

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



Require: σ

In this experiment, the first column of $\Delta \mathbf{X}$ 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 $\left(\Delta \mathbf{X}^T \Delta \mathbf{X} \right)^{-1}$ and horizontally $\frac{1}{\sigma}$.

Answer to exercise 59 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 $\begin{pmatrix} \overset{\Delta}{\mathbf{X}}^T & \overset{\Delta}{\mathbf{X}} \end{pmatrix}^{-1}$.
- 5: Let c be the greatest value of $\begin{pmatrix} \overset{\Delta}{\mathbf{X}}^T & \overset{\Delta}{\mathbf{X}} \end{pmatrix}^{-1}$.

Answer to exercise 59 III

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

$$\begin{aligned}1(b - [a_1, a_2] \cdot \mathbf{x} \geq 0) &= 1(0 - [a_1 - \frac{b}{2}, a_2] \cdot \mathbf{x} \geq 0) \\ &= 1(b - 2a_1 - [0, a_2] \cdot \mathbf{x} \geq 0)\end{aligned}$$

This identity adds to the general property when $b > 0$,

$$1(b - [a_1, a_2] \cdot \mathbf{x} \geq 0) = 1(1 - [\frac{a_1}{b}, \frac{a_2}{b}] \cdot \mathbf{x} \geq 0)$$

- 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.

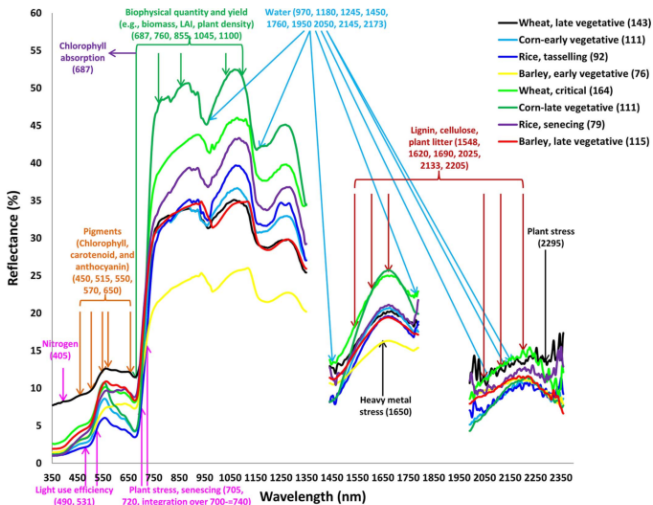


Fig. 8. Optimal hyperspectral narrowbands (HNBs). Current state of knowledge on hyperspectral narrowbands (HNBs) for agricultural and vegetation studies (inferred from [8]). The whole spectral analysis (WSA) using contiguous bands allow for accurate retrieval of plant biophysical and biochemical quantities 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.

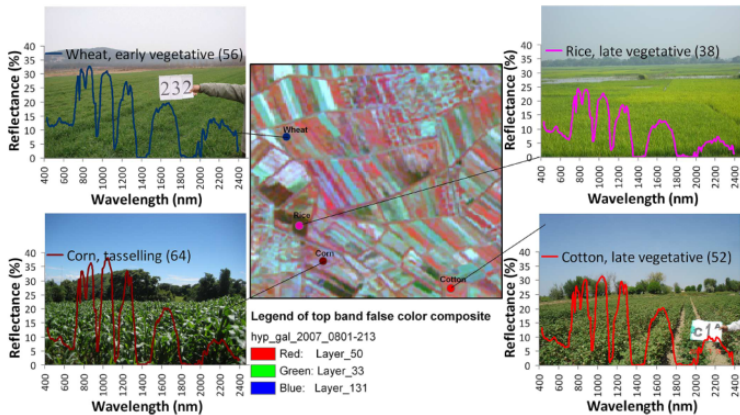


Fig. 3. Hyperion data of crops illustrated for typical growth stages in the Uzbekistan study area. The Hyperion data cube shown here is from a small portion of one of the two Hyperion images. The Hyperion spectra of crops are gathered from different farm fields in the two images and their average spectra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

There are many feature subsets

$$F = 10, F_1 = 5 \Rightarrow \binom{10}{5} = 252$$

$\text{prod}(1:10)/\text{prod}(1:5)/\text{prod}(1:5)$

- 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: \mathcal{F}

Ensure: \mathcal{F}'

- 1: **repeat**
- 2: Apply a 1-feature selection technique.
- 3: Update \mathcal{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:

$$\hat{f} = \arg \min_f |w_f|$$

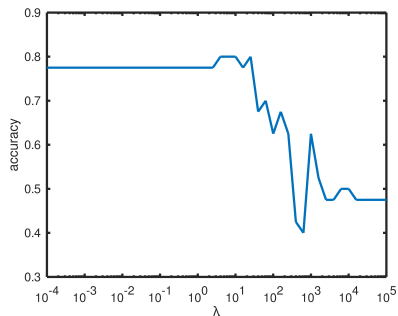
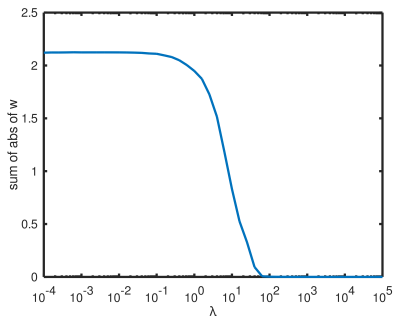
for a given λ .

LASSO:

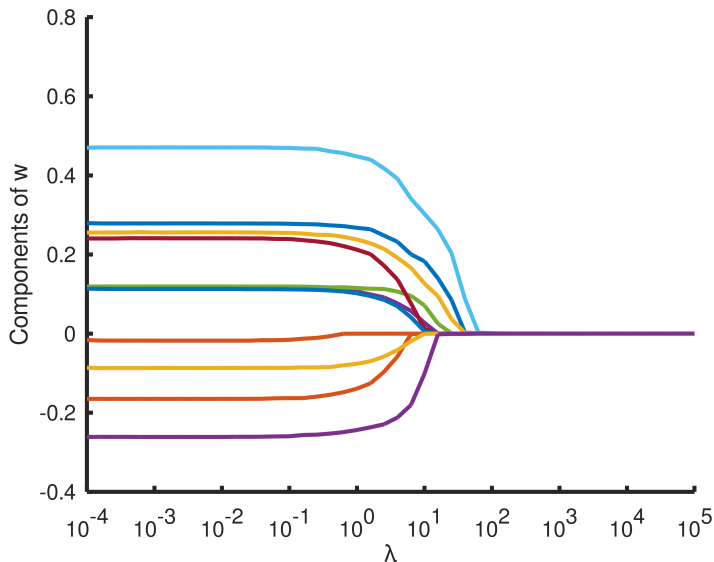
$$\hat{\lambda} = \arg \min_{\lambda} \{ \lambda \mid \exists f \mid \hat{w}_f = 0 \}$$

Lasso experiment

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



L1 minimization \Rightarrow features are cancelled



Conclusion of subsection 8, Feature selection

- 1 Classifying is not only a question of having the best accuracy. Explaining what happens is 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 I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

Table of Contents II

8. Supplementary material regarding matrices

Content of section 7, Spatial context I

- 7.1 Spatial context
- 7.2 Texture descriptors
- 7.3 Noise estimation
- 7.4 Spatial prior

- 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)



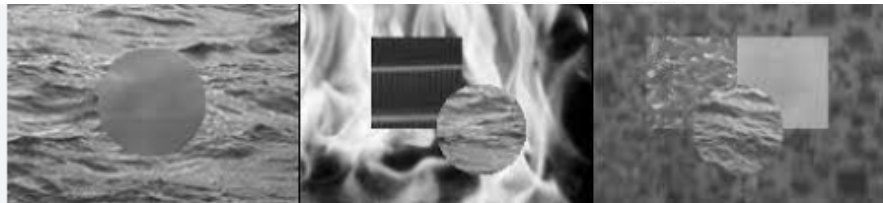
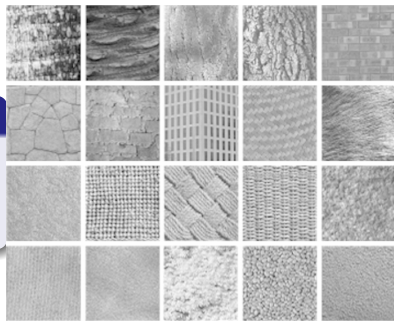
Content of section 7, Spatial context I

- 7.1 Spatial context
- 7.2 Texture descriptors
- 7.3 Noise estimation
- 7.4 Spatial prior

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

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

- 1 Horizontal filter

$$f'_{mn} = \frac{1}{5} \sum_{m'=m-2}^{m+2} f_{m'n}$$

- 2 Variance

$$f'_{mn} = \sum_{\mathcal{V}_{mn}} (f_{m'n'} - \mu_{mn})^2$$

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

- 3 Diversity index

$$f'_{mn} = \sum_g h(g)^2 \text{ where } h(g) \text{ is the estimated probability distribution}$$

Proposed techniques II

4 Correlation

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

5 Mean

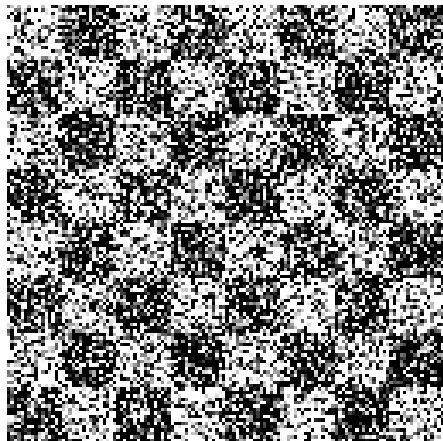
$$f'_{mn} = \frac{1}{25} \sum_{\psi_{mn}} f_{m'n'}$$

Exercise 60

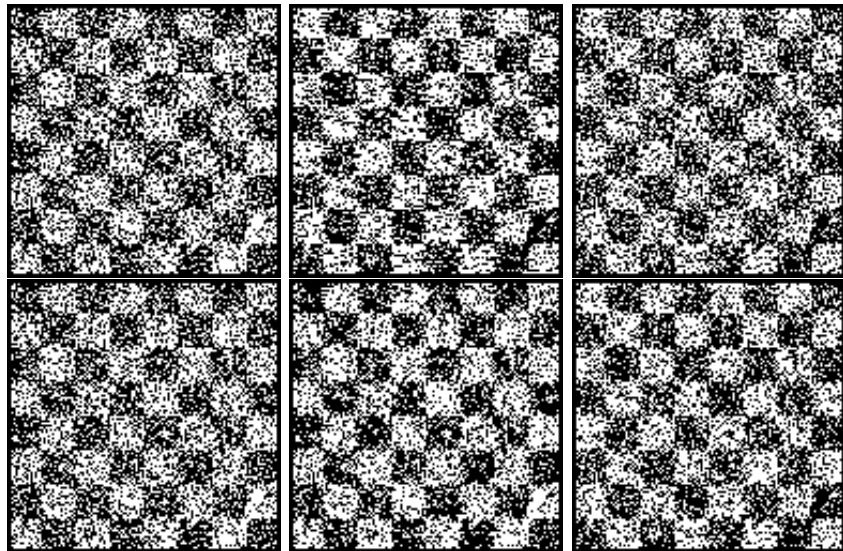
Considering a noisy image of a chessboard with only one feature.

1 Which technique has which property?

Application to a chessboard



Feature used in the kmeans algorithm in the next slide

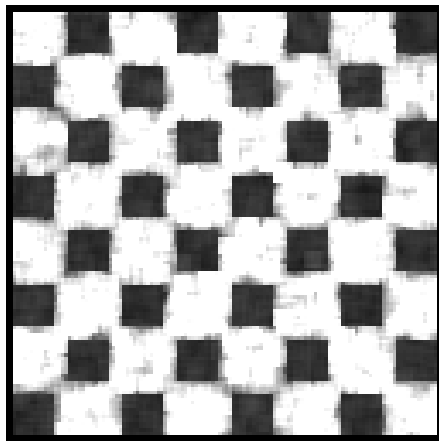
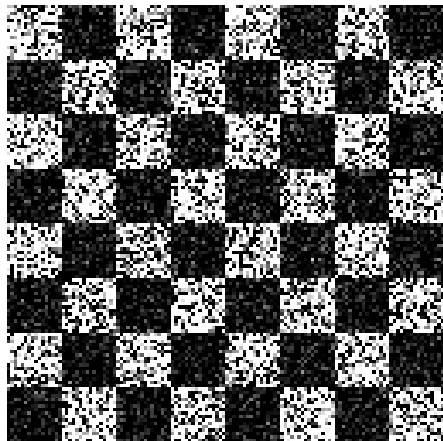


- Pixel value
- Diversity
- Horizontal filtering
- Correlation
- Variance
- Mean

Content of section 7, Spatial context I

- 7.1 Spatial context
- 7.2 Texture descriptors
- 7.3 Noise estimation**
- 7.4 Spatial prior

An example



Explaining the experiment

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

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

$$\text{noise}_{mn} = \sqrt{\sum_{\mathcal{V}_{mn}} (f_{m'n'} - \mu_{mn})^2}$$
$$\mu_{mn} = \frac{1}{25} \sum_{\mathcal{V}_{mn}} f_{m'n'}$$

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

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}, \mathbf{eZ}^T\mathbf{Z}\mathbf{e}^T=1} \mathbf{eX}^T\mathbf{X}\mathbf{e}^T$$

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

$$\hat{\mathbf{e}} = \arg \max_{\mathbf{e}} \mathbf{eX}^T\mathbf{X}\mathbf{e}^T - \lambda\mathbf{eZ}^T\mathbf{Z}\mathbf{e}^T$$

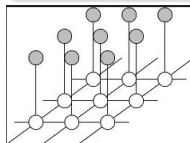
with λ chosen to fit the constraint.

Content of section 7, Spatial context I

- 7.1 Spatial context
- 7.2 Texture descriptors
- 7.3 Noise estimation
- 7.4 Spatial prior

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

$$y_{mn} = \text{chess Board and } x_{mn} \sim \mathcal{N}(y_{mn}, 2)$$

Equations

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

where

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

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

Finally we get a new global function to minimize

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

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

Table of Contents I

1. Classification of hyperspectral images
2. Image processing
3. Learning regarded as an optimization problem
4. Predicting the learning performances and probabilistic framework
5. More in depth with probabilities
6. Curse of dimensionality, regularization and sparsity
7. Spatial context

Table of Contents II

8. Supplementary material regarding matrices

Reordering a dataset

Exercise 61

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 61

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

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

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

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 $[\mathbf{x}_3^T, \mathbf{x}_1^T, \mathbf{x}_2^T]$

$$T \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \times 1 + \mathbf{1} \times 2 + 0 \times 3 \\ 0 \times 1 + 0 \times 2 + \mathbf{1} \times 3 \\ 1 \times 1 + 0 \times 2 + 0 \times 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

Content of section 8, Supplementary material regarding matrices I

8.1 Proving that kmeans is related to an optimization problem

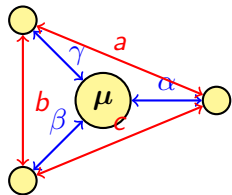
Star-triangle identity

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

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



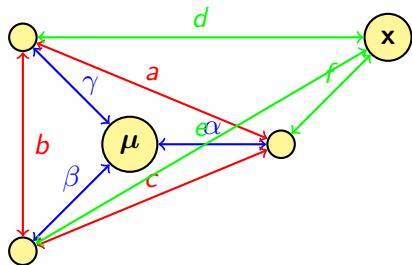
$$2 \times 3(\alpha^2 + \beta^2 + \gamma^2) = 2(a^2 + b^2 + c^2)$$

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} .

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



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

Exercise 62

We consider a dataset (\mathbf{X}, Y) and denote N_0, N_1, μ_0, μ_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

$$N_0\mu_0 + N_1\mu_1 = N\mu = \sum_{n=1}^N \mathbf{x}_n$$

where μ 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
$$N_0(Y') = N_0(Y) - 1, \quad N_1(Y') = N_1(Y) + 1,$$

Exercise

- 3 Let μ_0, μ_1 be the means of the 0 and 1-labeled samples before the modification. Let μ'_0, μ'_1 be the corresponding means after the modification. Show that

$$\mu'_0 - \mathbf{x} = \frac{N_0}{N_0 - 1}(\mu_0 - \mathbf{x})$$

- 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

$$J' - J = \frac{N_1}{N_1 + 1} \|\mu_1 - \mathbf{x}\|^2 - \frac{N_0}{N_0 - 1} \|\mu_0 - \mathbf{x}\|^2$$

- 5 Show that $J' \leq J$, when Y' is modified according to *kmeans*, still assuming that here only **one** component changes.

Answer to exercise 62 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

$$(N_0 - 1)(\boldsymbol{\mu}'_0 - \mathbf{x}) = (N_0 - 1)\boldsymbol{\mu}'_0 - (N_0 - 1)\mathbf{x}$$

$$= N_0\boldsymbol{\mu}_0 - \mathbf{x} - (N_0 - 1)\mathbf{x} = N_0(\boldsymbol{\mu} - \mathbf{x})$$

and hence that $\boldsymbol{\mu}'_0 - \mathbf{x} = \frac{N_0}{N_0 - 1}(\boldsymbol{\mu} - \mathbf{x})$

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

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

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

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

and

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

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

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

Answer to exercise 62 III

The definition of J and J' tells us

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

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

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

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