UNIVERSIDAD DE BURGOS

Sistemas Inteligentes
Curso 2023-2024

Reconocimiento de patrones. Fundamentos.

Curso 2024-2025

Pedro Latorre Carmona
Depto. Ingeniería Informática – UBU

Outline

- Introduction
- Classifiers based on Bayes decision theory
- Support Vector Machines (SVMs)
- Clustering
- *Houston*, we have a few problems!

Introduction

Introduction

How can we define what **pattern recognition** is?

From Wikipedia:

"**Pattern recognition** is a branch of **machine learning** that focuses on the recognition of patterns and regularities in data, **although it is in some cases considered to be nearly synonymous with machine learning**.

Pattern recognition systems are in many cases trained from labeled ***training*** data (supervised learning), but when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning).

The terms **pattern recognition**, **machine learning**, **data mining** and **knowledge discovery in databases** (KDD) are hard to separate…".

Introduction

What is a **pattern**?

A series of measurements whose combination will define the “object under study”.

What are the steps in a “pattern recognition” system?

We can consider the following ones:

Introduction

There are different *taxonomies* for Pattern recongition.

One of them is based on whether we have information about the class the data samples belong to or not.

A pattern recognition method is called:

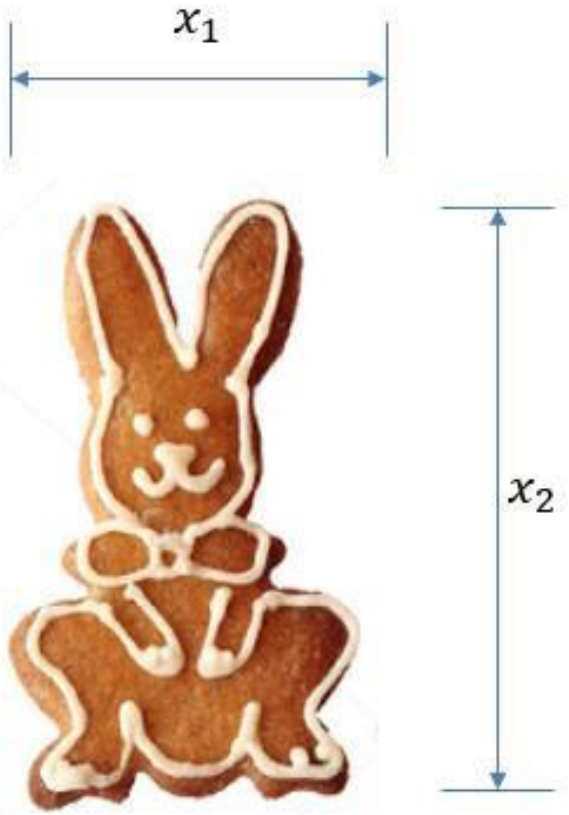
- *Supervised*: if we have access to the labels (class information) associated to the data samples.
- *Unsupervised*: if this information is not avalable.

Introduction

Imagine we have the following situation. We need to distinguish two types of animal crackers for food: *rabbits* from *sheeps*.

Introduction

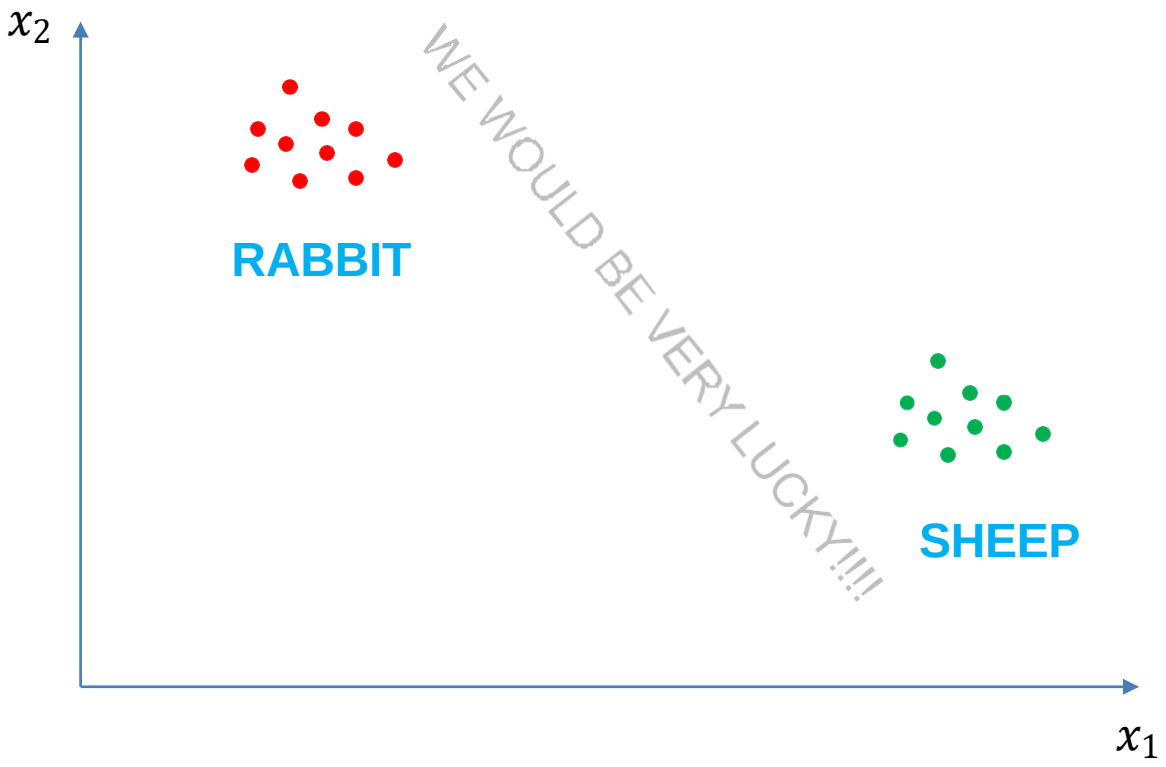
We can define the following two (simple) features:



and define $\boldsymbol{X} = (x_1, x_2)$ as a **feature vector**.

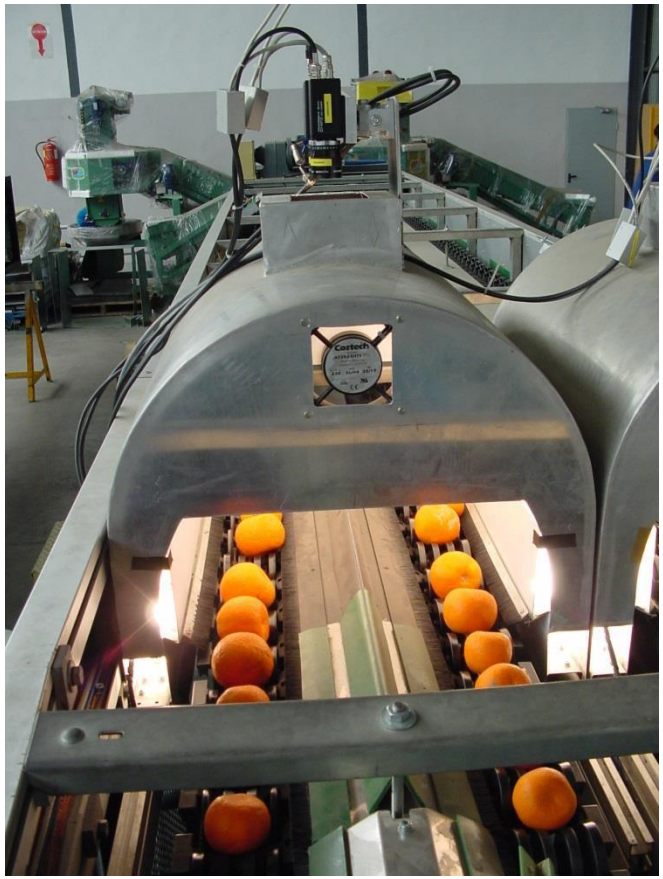
Introduction

What if we do the following plot and find…?



Introduction

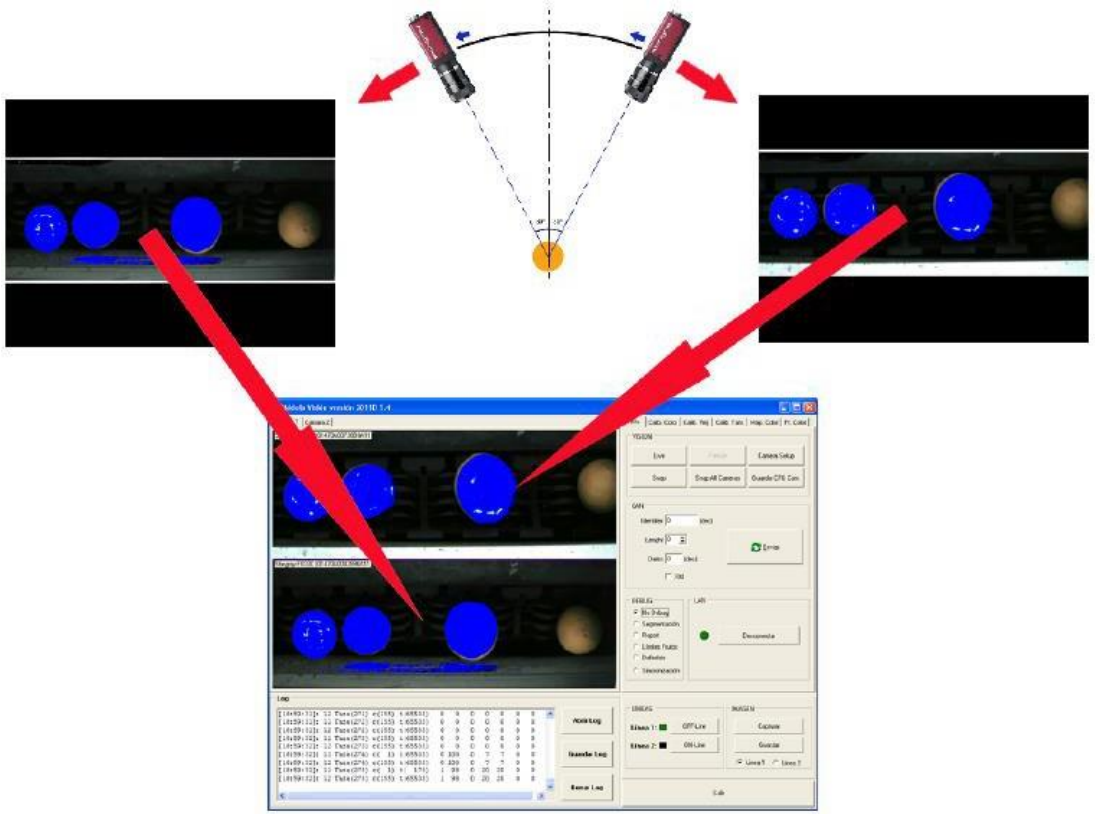
Imagine an on-line fruit inspection/sorting system:



Introduction

Imagine an on-line fruit inspection/sorting system:

We could “infer”
a *center* and
a *radius* of/in the fruit



The feature vector could be:

$$\boldsymbol{X_i} \equiv (\boldsymbol{C_i}, \boldsymbol{r_i});$$
$$C_i \equiv (x_i, y_i)$$

Introduction

A *supervised* pattern recognition dataset is formed by (X_i, y_i) pairs, where:

$X_i \equiv \{x_{1i}, x_{2i}, x_{3i}, \dots, x_{\boldsymbol{d}i}\}$;
$y_i \equiv$ Class the data sample i belongs to.

where $\boldsymbol{d}$ represents the so-called problem **DIMENSIONALITY**.

Any pattern recognition dataset is divided into:

- A *Training* dataset
- A *Test* dataset

Introduction

A complete dataset must **ALWAYS** be divided into a *training* and a *test* dataset, and **size(*training_set*) > size(*test_set*)**

50 - 50

80 – 20
(70-30,…)

Hold out

Introduction

How can we measure classification accuracy?

$$\boldsymbol{X} \equiv \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots, \boldsymbol{x}_N\}; \quad Y = \{y_1, y_2, \dots, y_N\} \in \{+1, -1\}$$

	"Illness"	No "illness"
Classified as "Illness"	True Positives (TP)	False Positives (FP)
Classified as "No illness"	False Negatives (FN)	True Negatives (TN)

$$\boldsymbol{Acc}\ (\%) = \left(\frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+TN+FP+FN}}\right) \cdot \boldsymbol{100}$$

Introduction

What if there are more than **2** classes?

$$\boldsymbol{X} \equiv \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots, \boldsymbol{x}_N\}; \quad Y = \{y_1, y_2, \dots, y_N\} \in \{1, 2, 3, \dots\}$$

$i = 1$ $\quad$ $i = 2$ $\quad$ $i = 3$ $\quad$ … $\quad$ $i = l$

1 $\quad$ **2 3 4 5 6 …** $\quad$ **2** $\quad$ **1 3 4 5 6 …** $\quad$ **3** $\quad$ **1 2 4 5 6**

$$\overline{Acc} = \frac{\sum_{i=1}^{l} \left(\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \right)}{l}$$

Supervised classification (SC)

SC- Bayes decision rules

Classifiers based on Bayes *decision theory*

SC- Bayes decision rules

Let's imagine the following situation:

A school is formed by 60% boys and 40% girls. We randomly pick up a student. What is the probability we chose a boy?

Answer: $\boldsymbol{p(Boy) \equiv p(\omega = \omega_1) = 0.6}$

SC- Bayes decision rules

But….What if….?

We are told all boys (100%) wear pants and half girls (50%) wear pants and half (50%) wear skirts. A camera sees a student with pants, what is the probability that the student is a boy?

$$p(Boy|Pants) \equiv p(\omega_1|x); \quad p(Pants|Boy) \equiv p(x|\omega_1)$$

$$p(Boy|Pants) = \frac{p(Boy)\cdot p(Pants|Boy)}{p(Boy)\cdot p(Pants|Boy)+p(Girl)\cdot p(Pants|Girl)} = \frac{0.6*1}{(0.6*1)+(0.4*0.5)} = \mathbf{0.75}$$

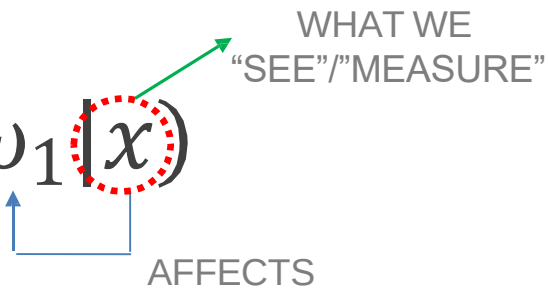
(Before): $p(Boy) \equiv p(\omega_1) = 0.6$

FURTHER information changes our ability to better estimate probabilities.

SC- Bayes decision rules

BEFORE (ANY MEASUREMENT): $p(\omega_1)$

AFTER (ANY MEASUREMENT): $p(\omega_1|x)$



Therefore, Bayes rule is **COOL**!

SC- Bayes decision rules

Let us define a variable considering a label associated to each class (rabbit, sheep).

Rabbit: $\omega = \omega_1$

Sheep: $\omega = \omega_2$

The ***a priori probability*** reflects our knowledge about how likely are classes to appear, **BEFORE** any measurement is taken.

These probabilities are:

$p(\omega_1);\ p(\omega_2)$

SC- Bayes decision rules

In general, for ***C*** classes, we would have:

$$\sum_{i=1}^{C} p(\omega_i) = 1$$

What do you think about the following plan (2 classes for now)?:

If $p(\omega_1) > p(\omega_2)$ → **Decide** that the **sample** belongs to **class ω_1**

If $p(\omega_1) \leq p(\omega_2)$ → **Decide** that the **sample** belongs to **class ω_2**

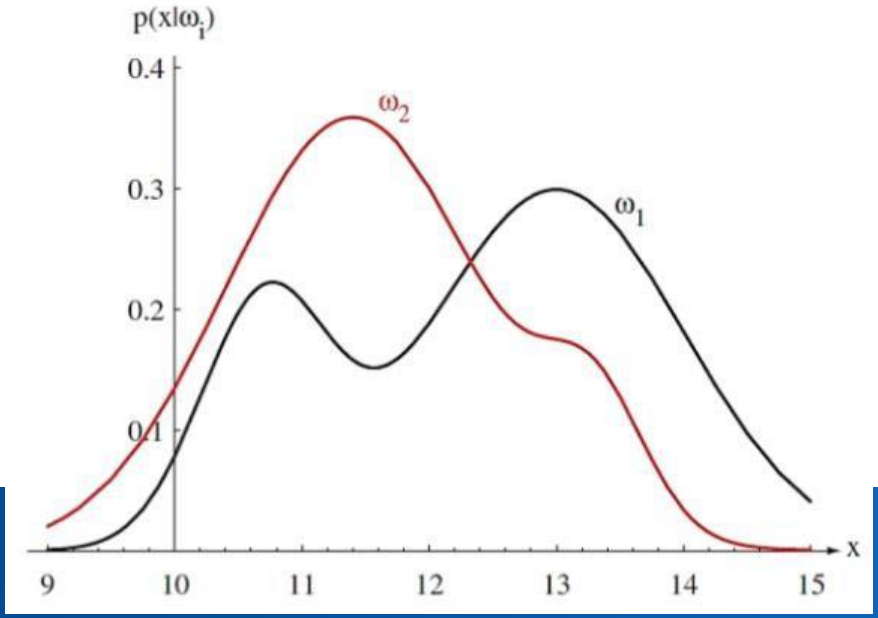
Answer: We would always be selecting the same class!!

SC- Bayes decision rules

Therefore, somehow, **we need to introduce the data into the model.**

A **feature** is a variable you can ***measure.*** A feature can be: (a) a scalar, $x \in \mathbb{R}$; (b) a vector, $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d}}$

The class-conditional probability density, $p(\boldsymbol{x}|\omega_i)$ reflects our assumptions about the **underlying distribution** for the data in the class.



SC- Bayes decision rules

What do we really need, then?

$\boldsymbol{p(\omega_i|x)}$

How can we obtain it? Using Bayes rule:

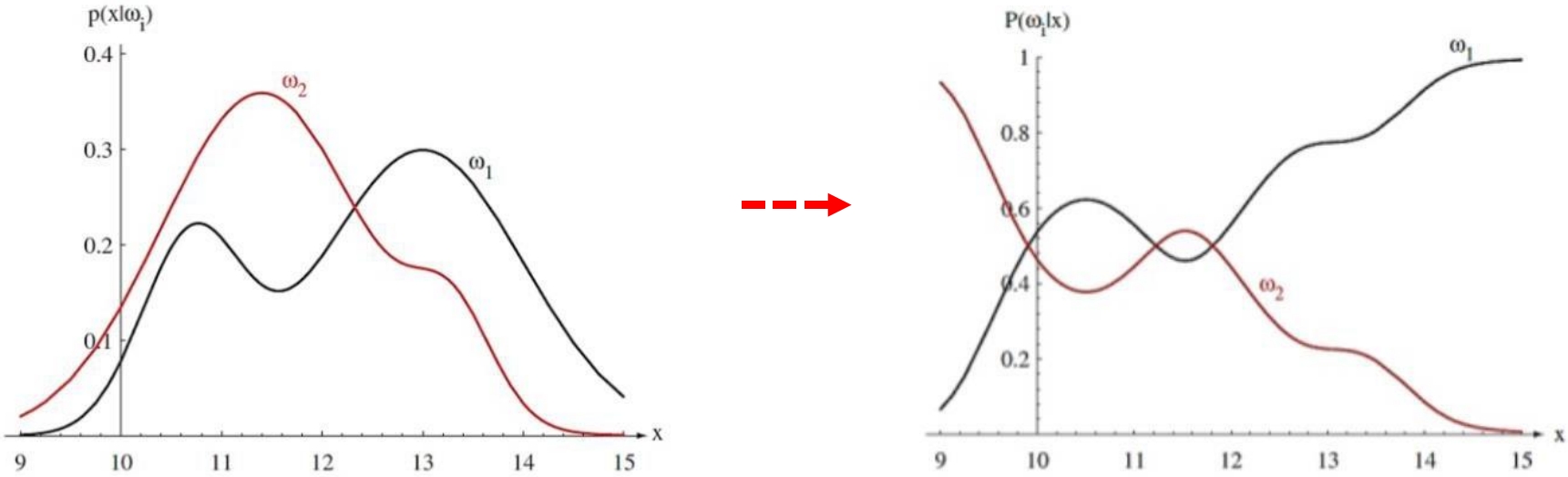
$$p(\omega_i, \boldsymbol{x}) = p(\boldsymbol{x}|\omega_i) \cdot p(\omega_i) = p(\omega_i|\boldsymbol{x}) \cdot p(\boldsymbol{x}) \Longrightarrow$$

$$\Longrightarrow \quad p(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i) \cdot p(\omega_i)}{p(\boldsymbol{x})} = \boxed{\frac{p(\boldsymbol{x}|\omega_i) \cdot p(\omega_i)}{\sum_{j=1}^{C} p(\boldsymbol{x}|\omega_j) \cdot p(\omega_j)} = p(\omega_i|\boldsymbol{x})}$$

SC- Bayes decision rules

Then:

$$p(\omega_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_i) \cdot p(\omega_i)}{\sum_{j=1}^{c} p(\boldsymbol{x}|\omega_j) \cdot p(\omega_j)}$$

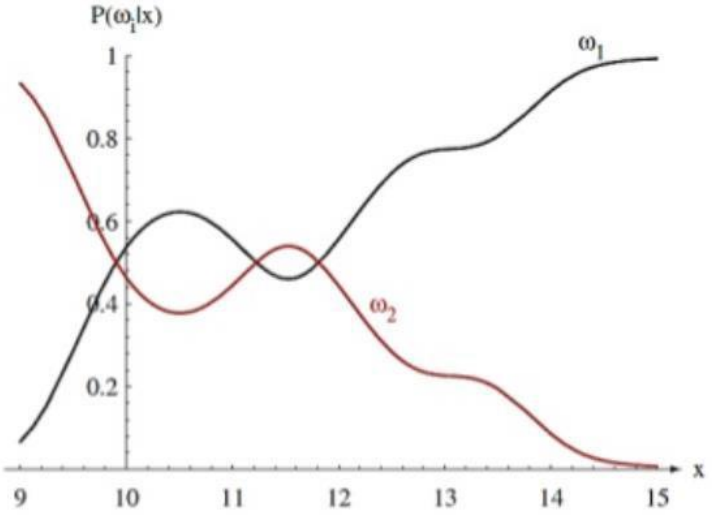


SC- Bayes decision rules

From the previous plot:

- If $p(\omega_1|\boldsymbol{x}) > p(\omega_2|\boldsymbol{x}) \rightarrow$ **Sample $\boldsymbol{x}$** belongs to **class $\boldsymbol{\omega_1}$**

- If $p(\omega_2|\boldsymbol{x}) > p(\omega_1|\boldsymbol{x}) \rightarrow$ **Sample $\boldsymbol{x}$** belongs to **class $\boldsymbol{\omega_2}$**



SC- Bayes decision rules

Therefore, the class would be:

$$\omega^*(\boldsymbol{x}) = \arg\max_i [p(\omega_i|\boldsymbol{x})]$$

Maximum A Posteriori (**MAP**) criterion

The probability of error (for the two class situation) would be:

$$p(\varepsilon|\boldsymbol{x}) = \begin{matrix} p(\omega_1|\boldsymbol{x}) & if\ we\ decide\ \omega_2 \\ p(\omega_2|\boldsymbol{x}) & if\ we\ decide\ \omega_1 \end{matrix}$$

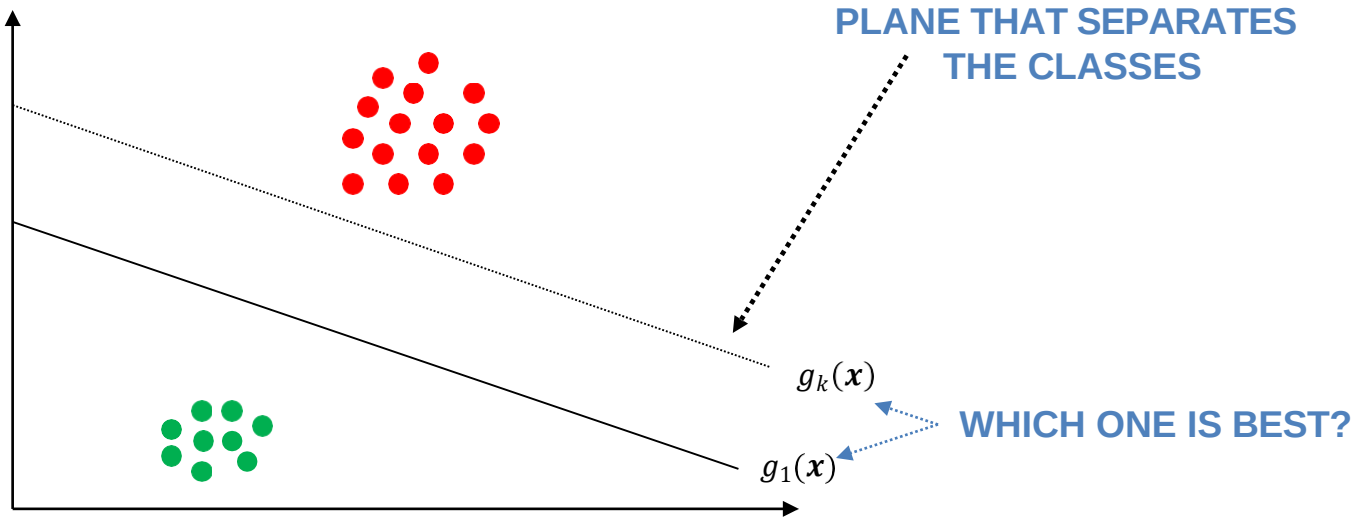
Support Vector Machines (SVMs)

SC (Linear)-SVMs

Let us consider a group of points $\{\boldsymbol{x}_i, i = 1, \ldots, N\}$ that belong to one of two classes $\{C_1, C_2\}$ assumed to be linearly separable by the hyperplane:

$$g(\boldsymbol{x}) \equiv \boldsymbol{\omega}^T \boldsymbol{x} + \omega_0 = 0$$

We may have many solutions $\{g_1(\boldsymbol{x}), \ldots, g_k(\boldsymbol{x})\}$



SC (Linear)-SVMs

The distance between a point and a line should be given by:

In the case of a line in the plane given by the equation $ax + by + c = 0$, where a, b and c are real constants with a and b not both zero, the distance from the line to a point (x_0, y_0) is[1][2]:p.14

$$\text{distance}(ax + by + c = 0, (x_0, y_0)) = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}.$$

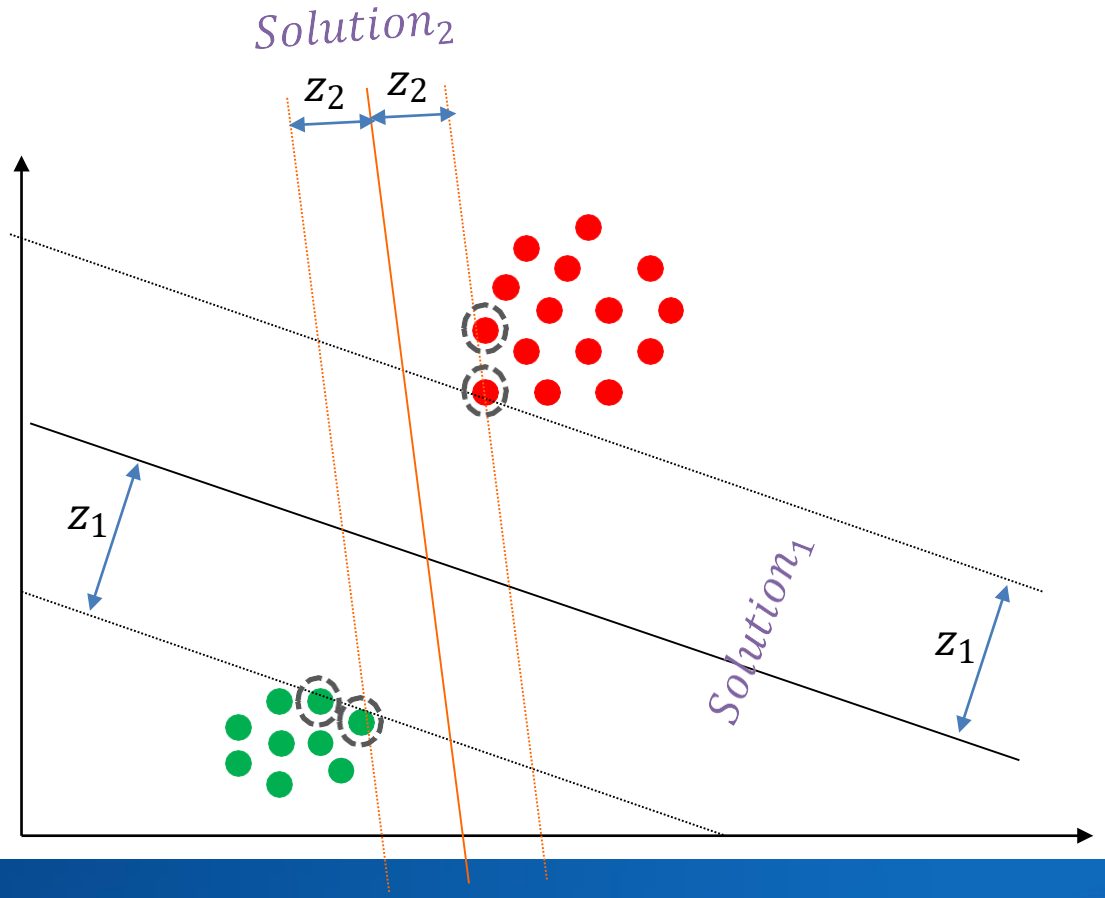
https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line#:~:text=In%20Euclidean%20geometry%2C%20the%20distance,nearest%20point%20on%20the%20line.

SC (Linear)-SVMs

Our aim would be to obtain $g(x) \equiv \boldsymbol{\omega}^T \boldsymbol{x} + \omega_0$ that maximizes z:

$\boldsymbol{\omega} = (\omega_1, \omega_2)$

$z = \frac{|g(\boldsymbol{x})|}{\|\boldsymbol{\omega}\|} = \frac{|g(\boldsymbol{x})|}{\sqrt{\omega_1^2 + \omega_2^2}}$



SC (Linear)-SVMs

This would be equivalent to:

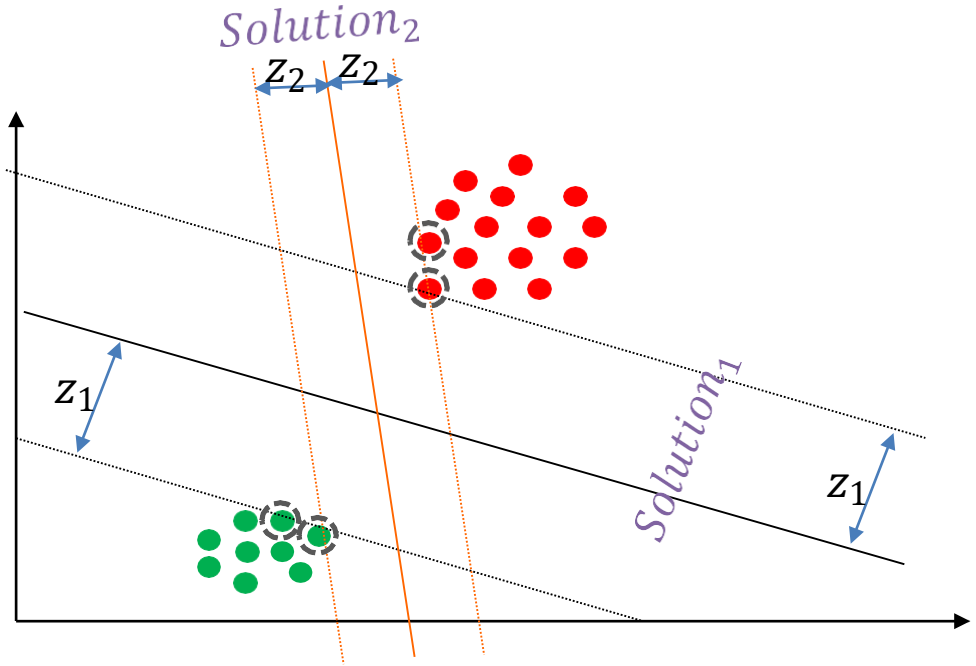
$maximize\ \left(\frac{2}{\|\boldsymbol{\omega}\|}\right)$

Or

$minimize\left(\frac{1}{2}\|\boldsymbol{\omega}\|^2\right)$

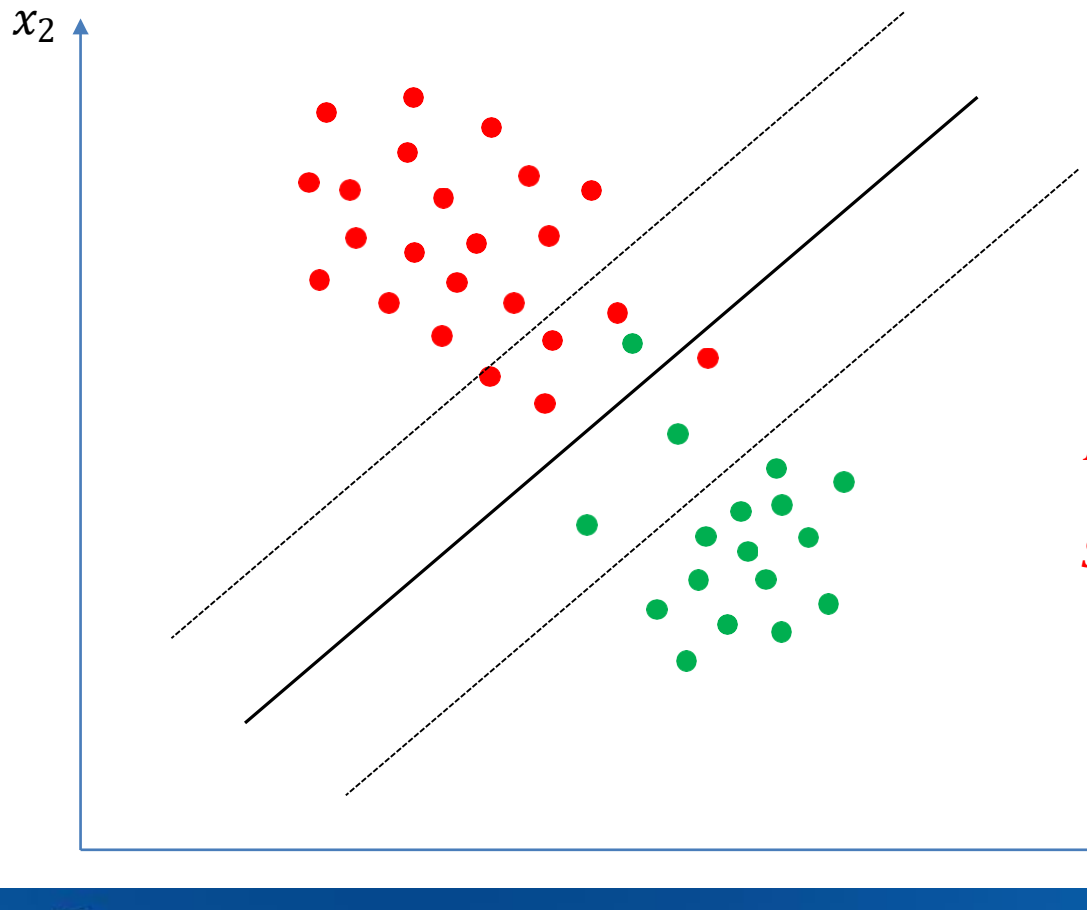
$subject\ to$:

$$\boldsymbol{\omega}^T\boldsymbol{x} + \omega_0 \geq 1 \quad \forall \boldsymbol{x} \in C_1$$
$$\boldsymbol{\omega}^T\boldsymbol{x} + \omega_0 \leq -1 \quad \forall \boldsymbol{x} \in C_2$$



SC (Linear)-SVMs

What if we have unseparable classes? We allow for some (N) errors!



$$\textit{Minimize:} \quad \left[\tfrac{1}{2}\|\boldsymbol{\omega}\|^2 + \mathrm{C} \cdot \sum_{i=1}^{N} \xi_i\right]$$

$$\textit{Subject to:} \quad \boldsymbol{\omega}^T \boldsymbol{x}_i + \omega_0 \geq 1 - \xi_i$$

$$\boldsymbol{\omega}^T \boldsymbol{x}_i + \omega_0 \leq -1 + \xi_i$$

SC (Linear)-SVMs

It can be shown that:

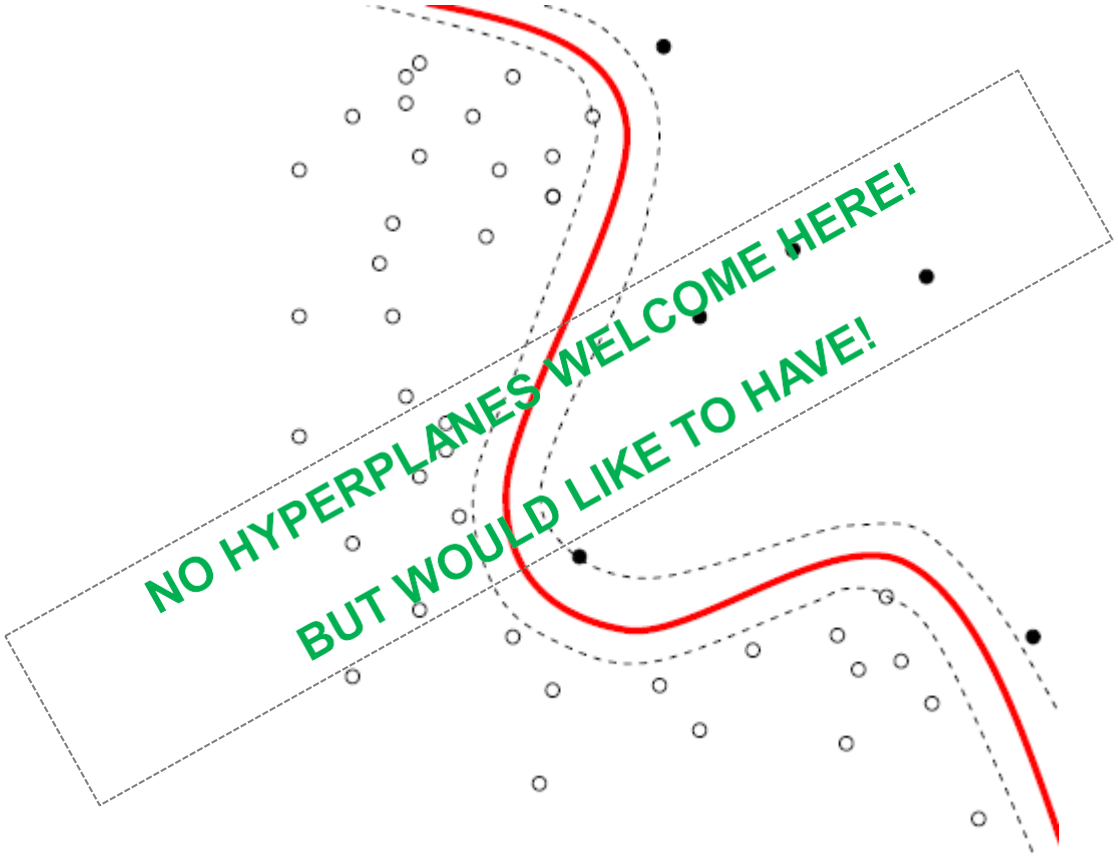
$$\boldsymbol{\omega} = \sum_{i=1}^{N_{SV}} \lambda_i \cdot y_i \cdot \boldsymbol{x}_i$$

where: $y_i \in \{-1, +1\}$ and N_{SV} is the number of training samples that define the hyperplane slope.

This is a CAPITAL result because it says the solution is a linear combination of some training samples, called SUPPORT VECTORS (SV) (and therefore the NAME!).

SC (Non linear)-SVMs

How could we achieve the following (needed by the data features)?



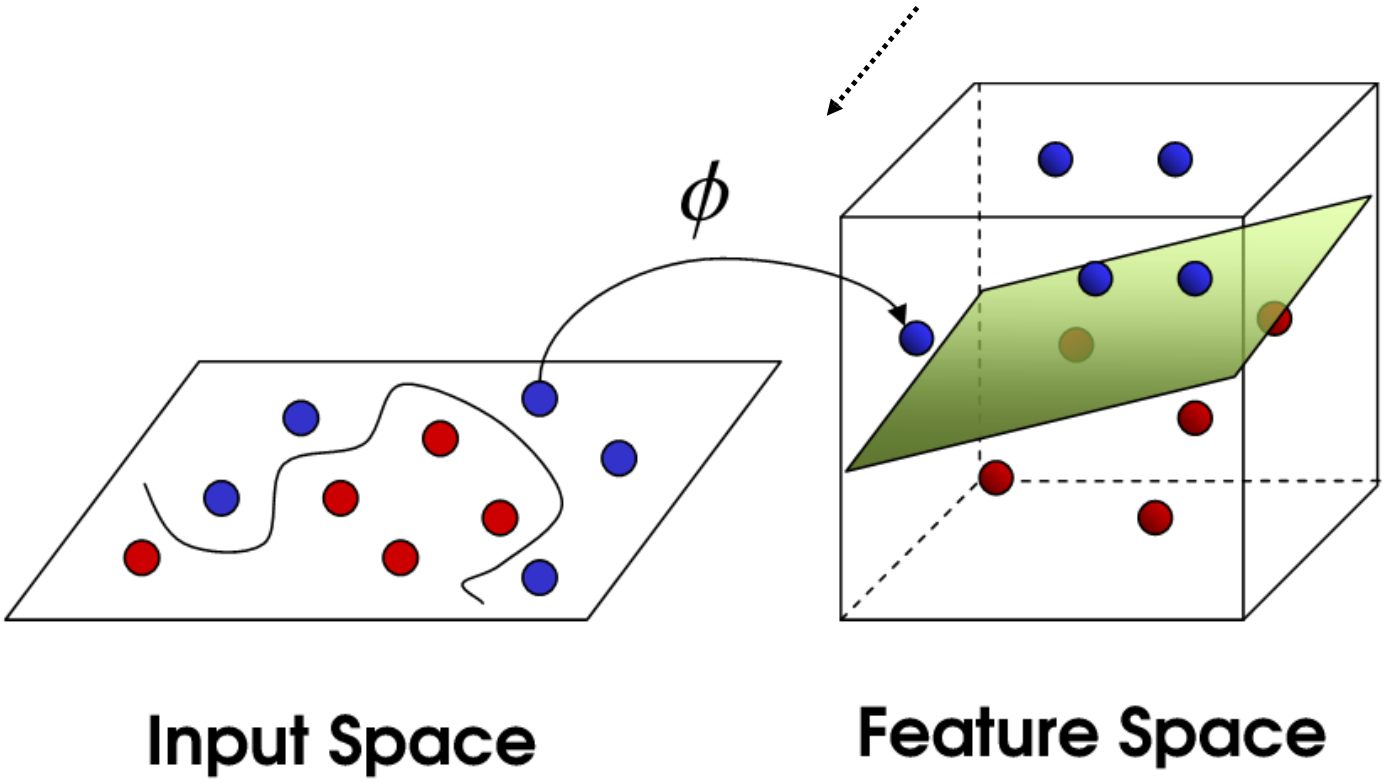
SC (Non linear)-SVMs

Vladimir Vapnik

(1936 -)

PhD Statistics, Moscow, 1964.

INCREASE THE NUMBER OF DIMENSIONS OF THE SPACE TO MAKE THE DATA POINTS LINEARLY SEPARABLE

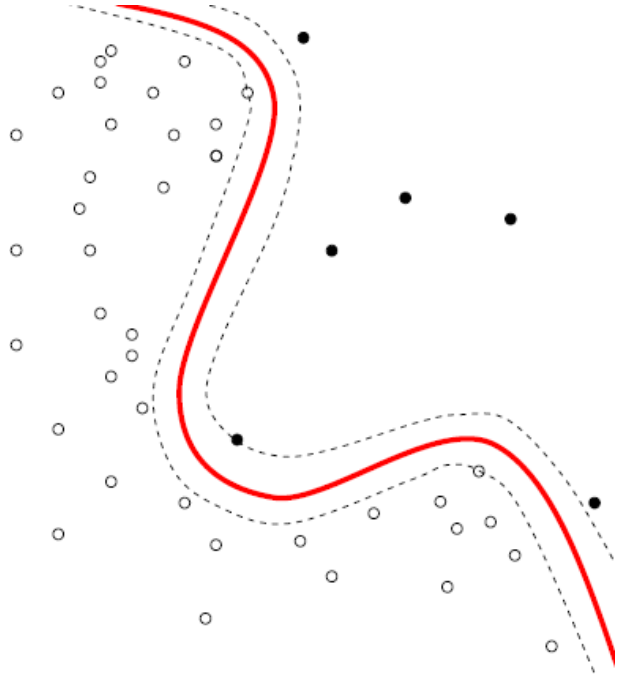


SC (Non linear)-SVMs

In a similar way it can be shown that:

$$g(\boldsymbol{x}) = \sum_{i=1}^{N_{SV}} \lambda_i \cdot y_i \cdot K(\boldsymbol{x}_i, \boldsymbol{x}) + \boldsymbol{\omega}_0$$

Kernel



where :

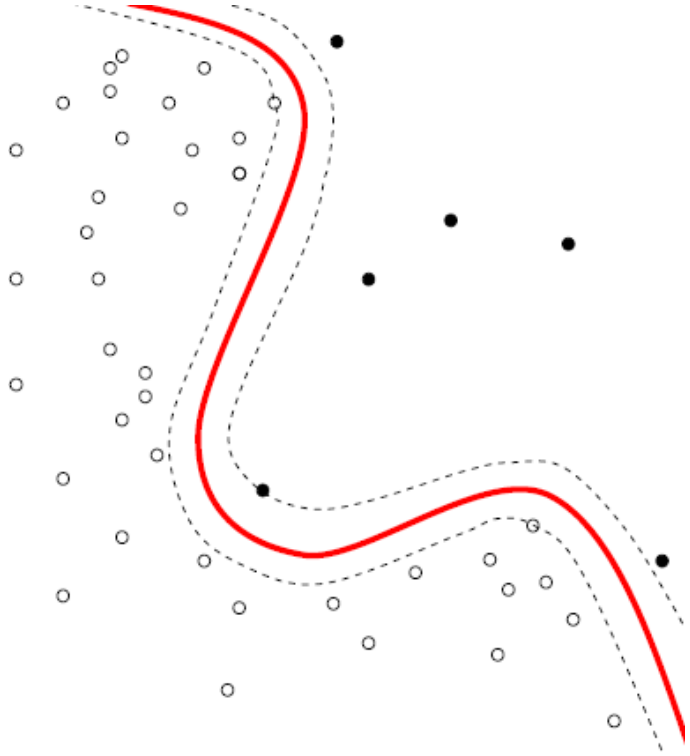
$y_i \in \{-1, +1\}$

N_{SV} is the number of training samples that the solution considers.

SC (Non linear)-SVMs

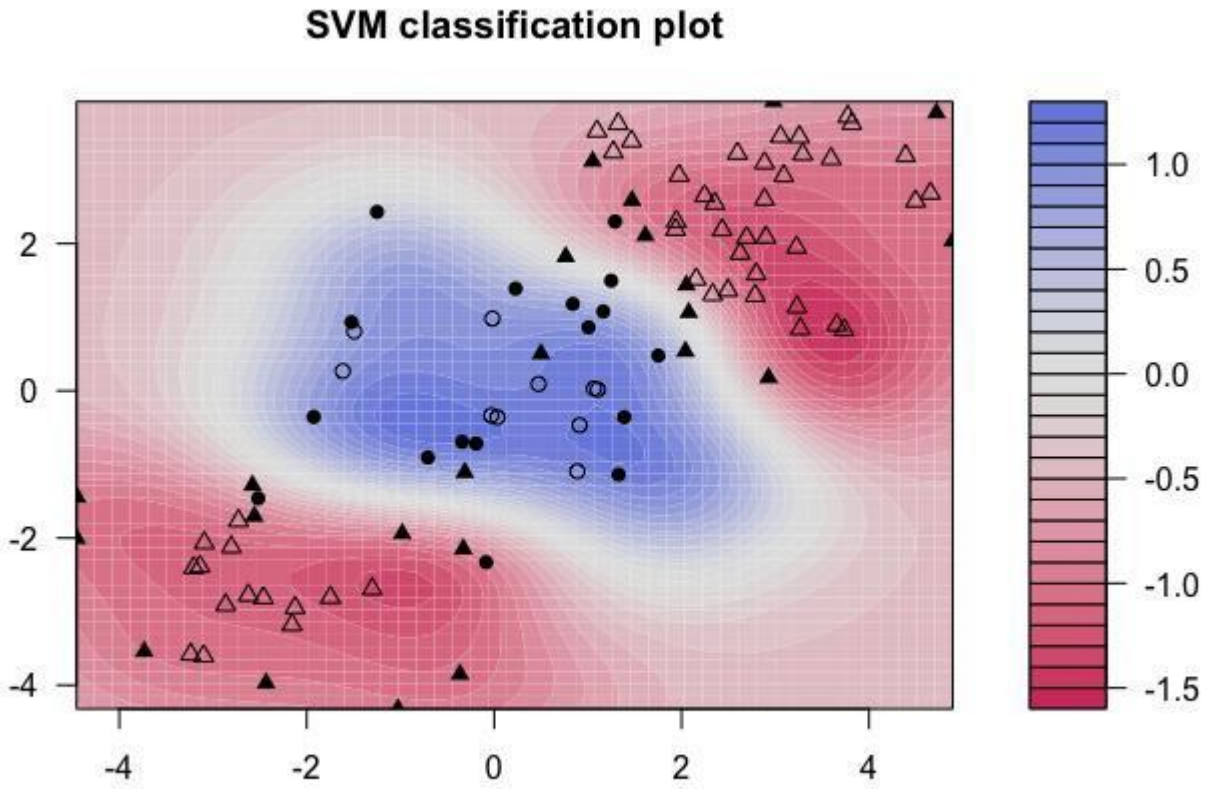
The most used Kernel is called "**Radial Basis Function**" (**RBF**)

$$K(\boldsymbol{x}_i, \boldsymbol{x}) \propto e^{-\frac{\|x_i - x\|^2}{\gamma^2}}$$

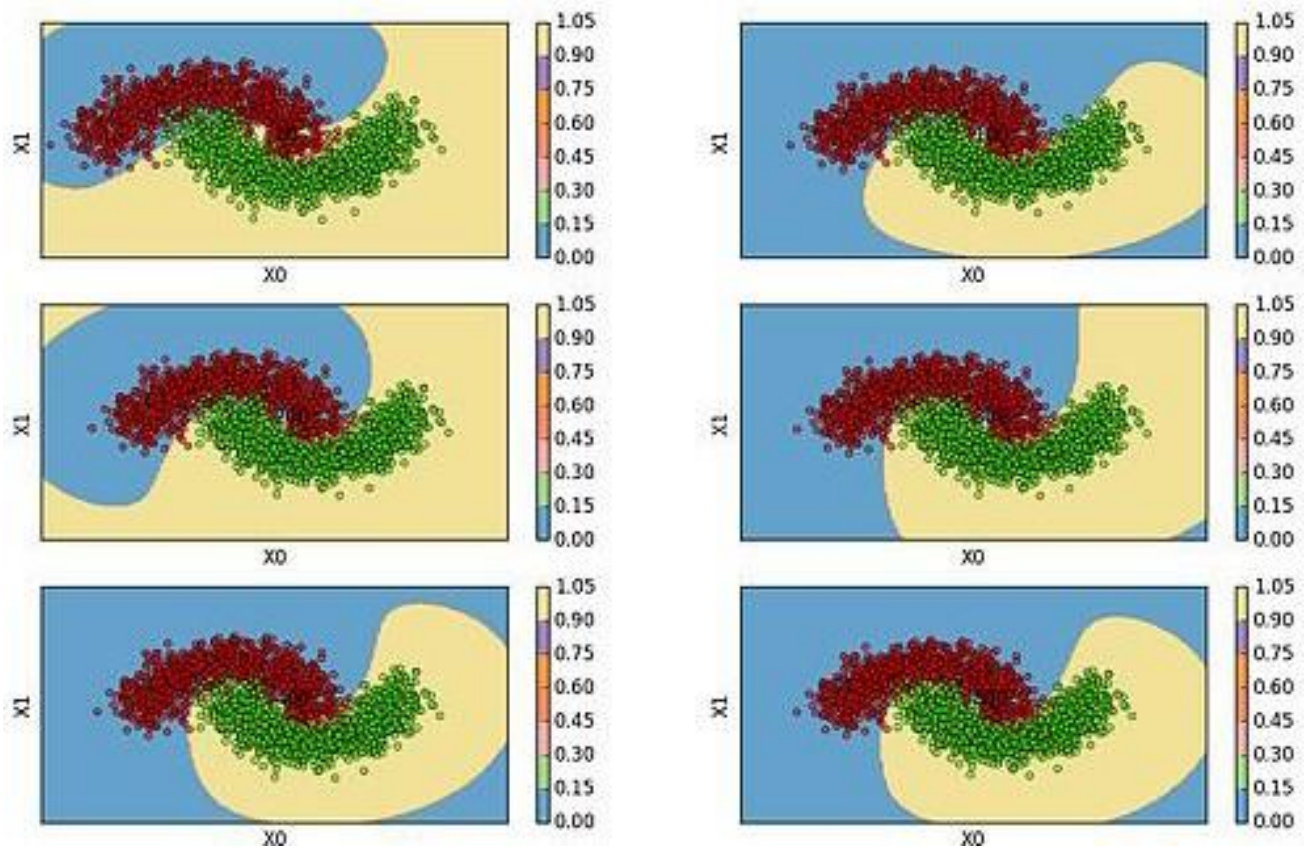


SC (Non linear)-SVMs

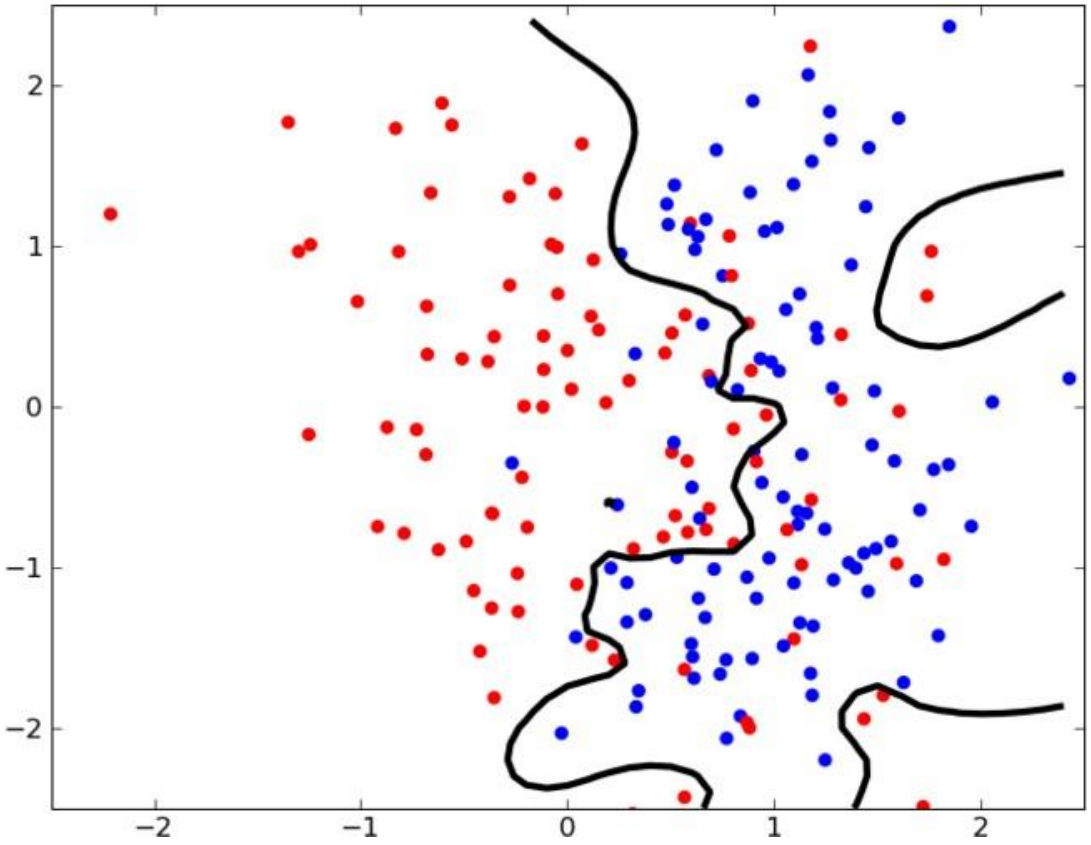
In general, the *border* can be very complicated, even *closed*.



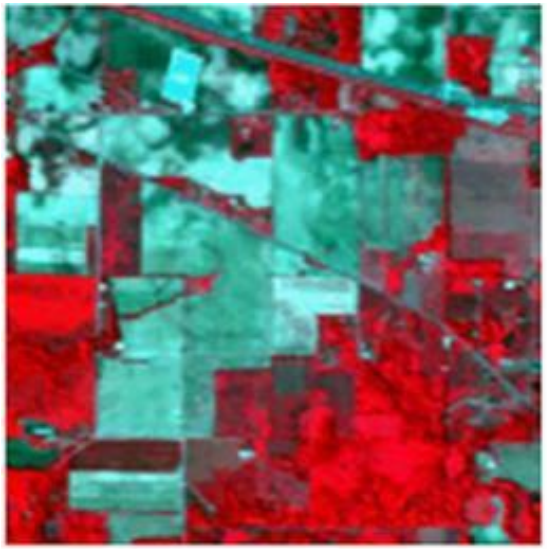
SC (Non linear)-SVMs



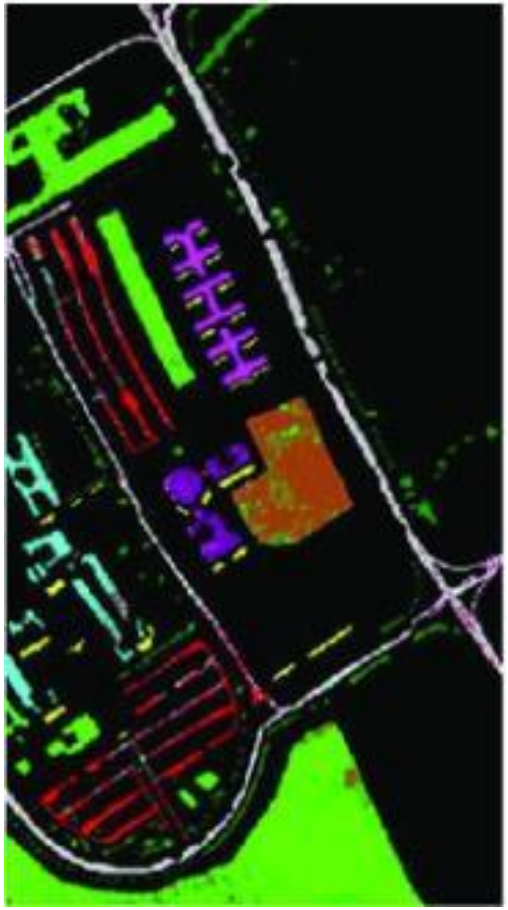
SC (Non linear)-SVMs



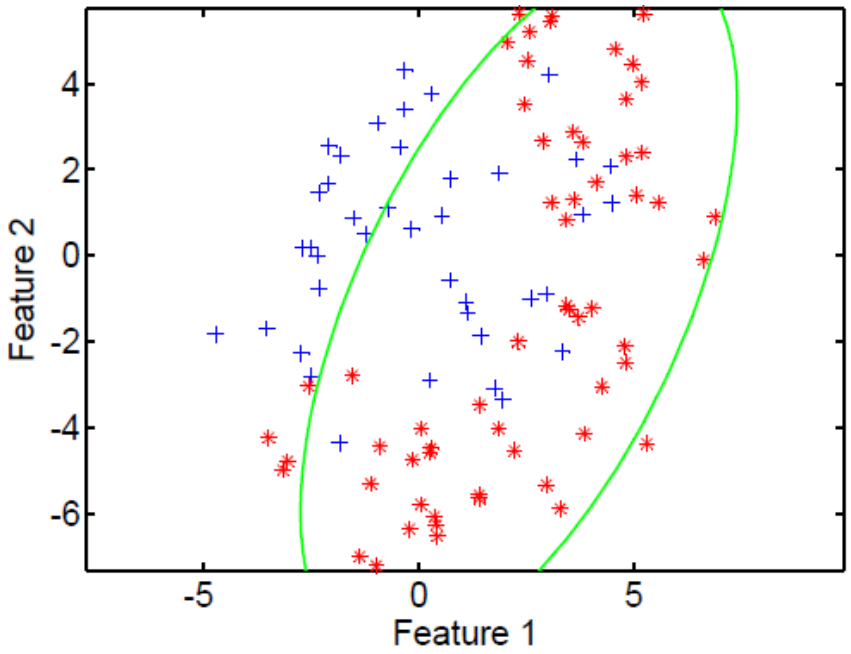
SC (Non linear)-SVMs

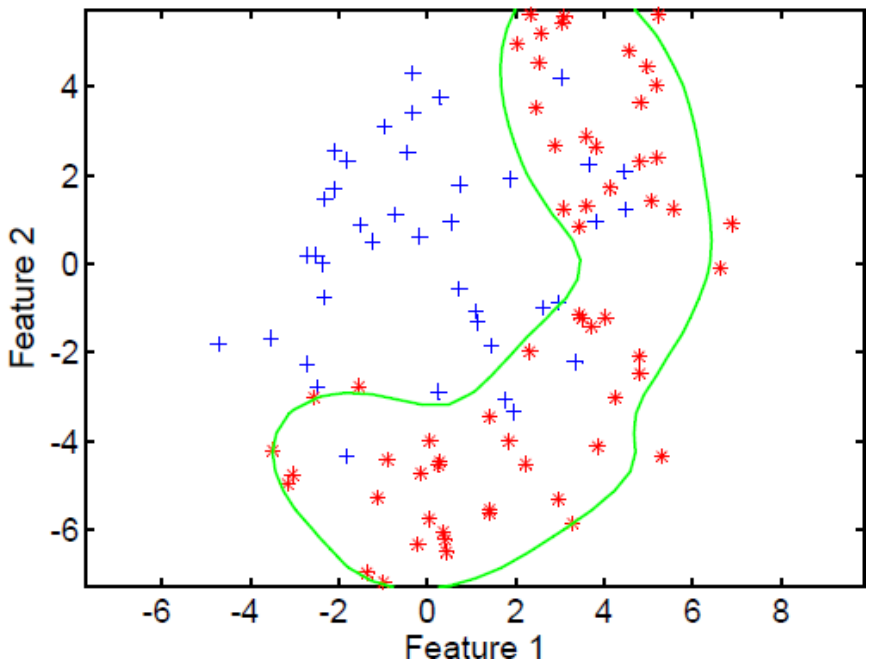


SC (Non linear)-SVMs



SC One class-SVM (OC-SVM)

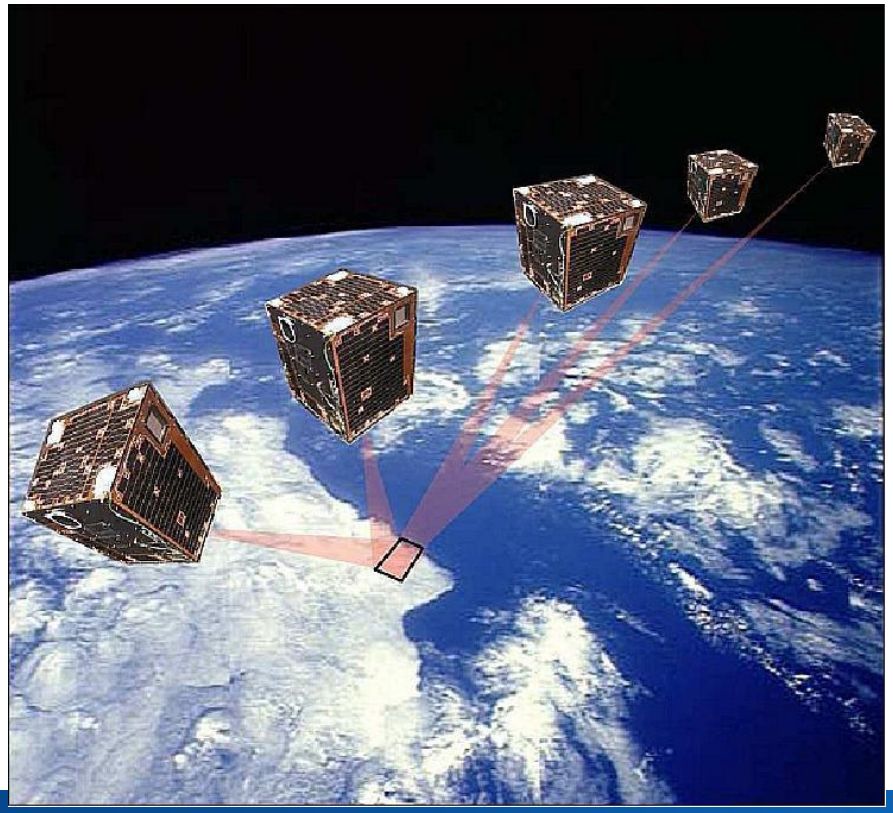




SC One class-SVM (OC-SVM)

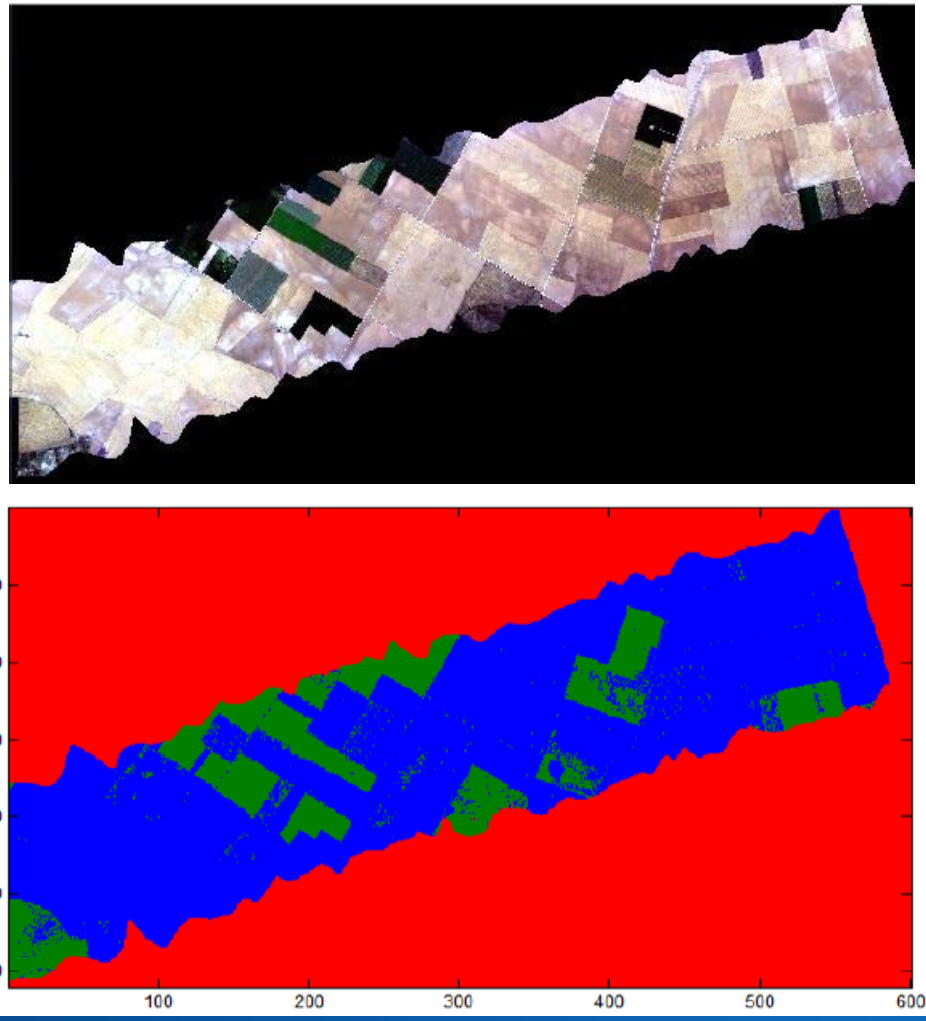
Compact High Resolution Imaging Spectrometer (**CHRIS**) on **PROBA** platform

Multispectral + Multiangular capabilities



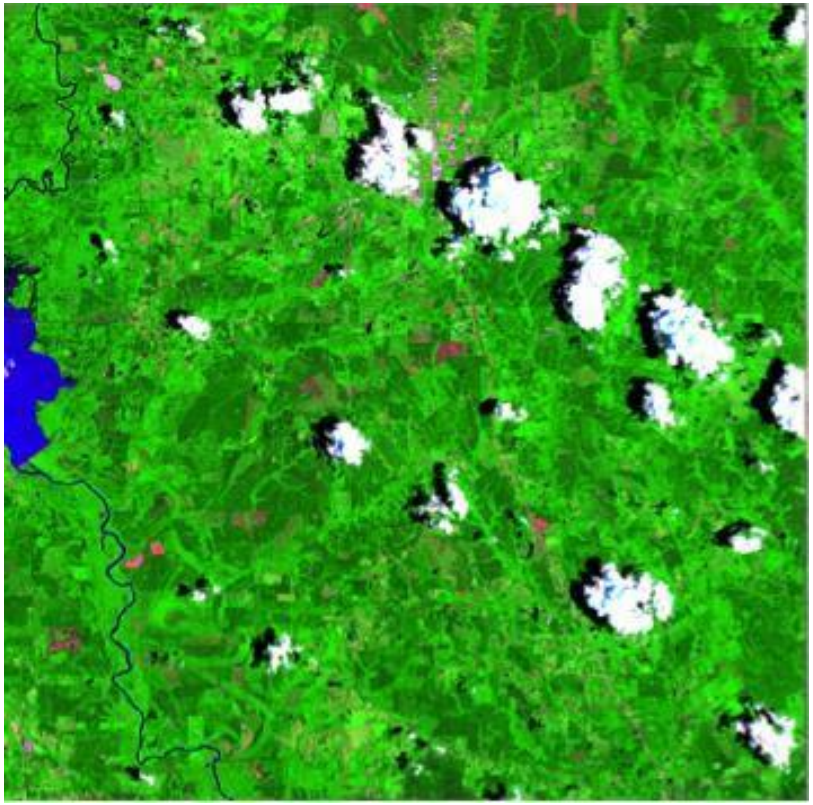
SC One class-SVM (OC-SVM)

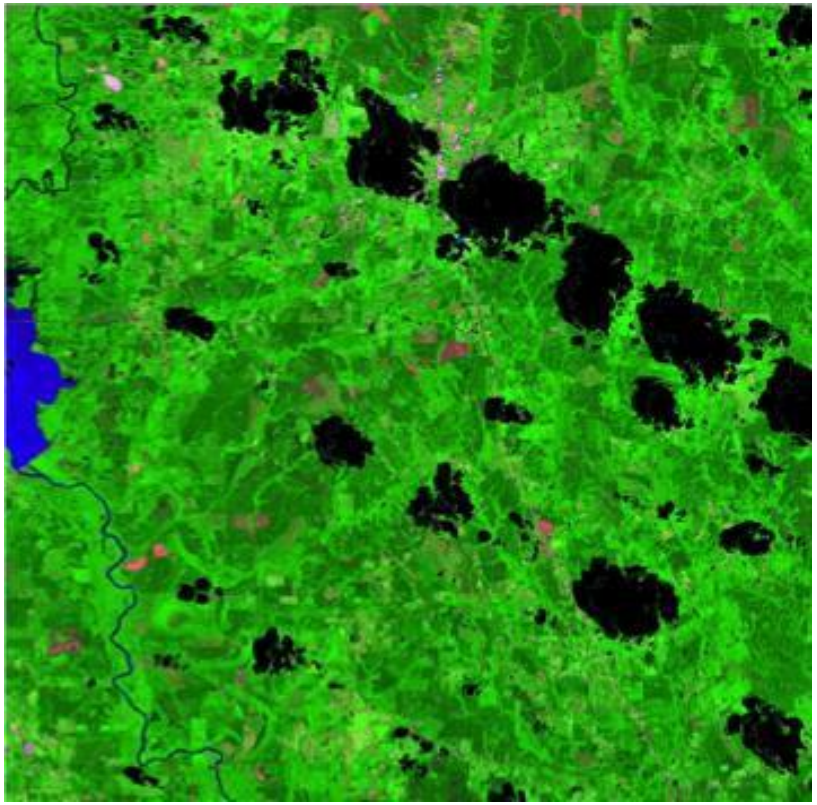
Class of interest:
Vegetation



SC One class-SVM (OC-SVM)

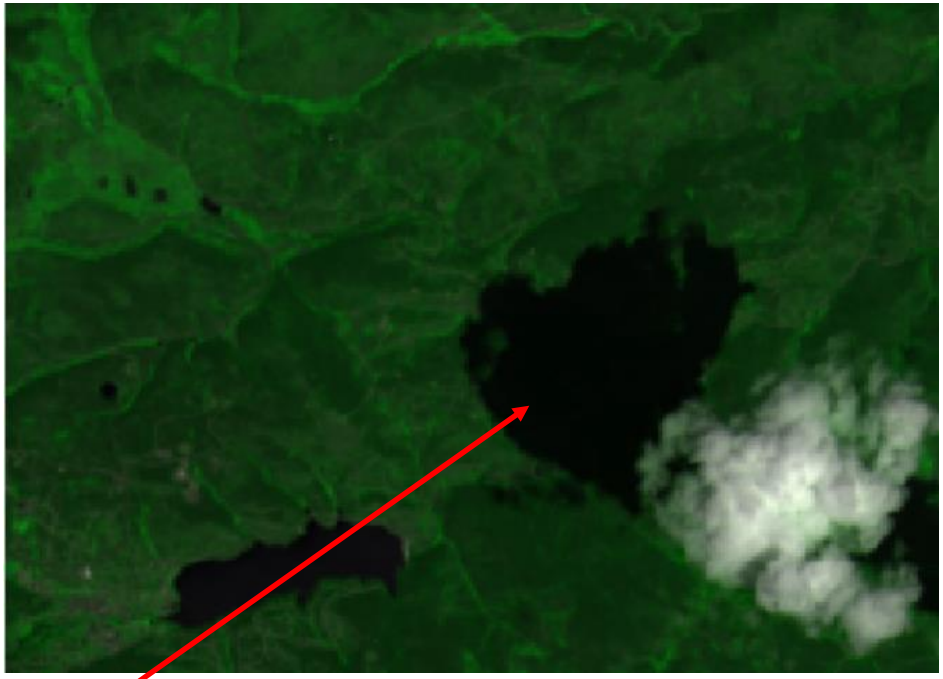
A cloud mask





SC One class-SVM (OC-SVM)

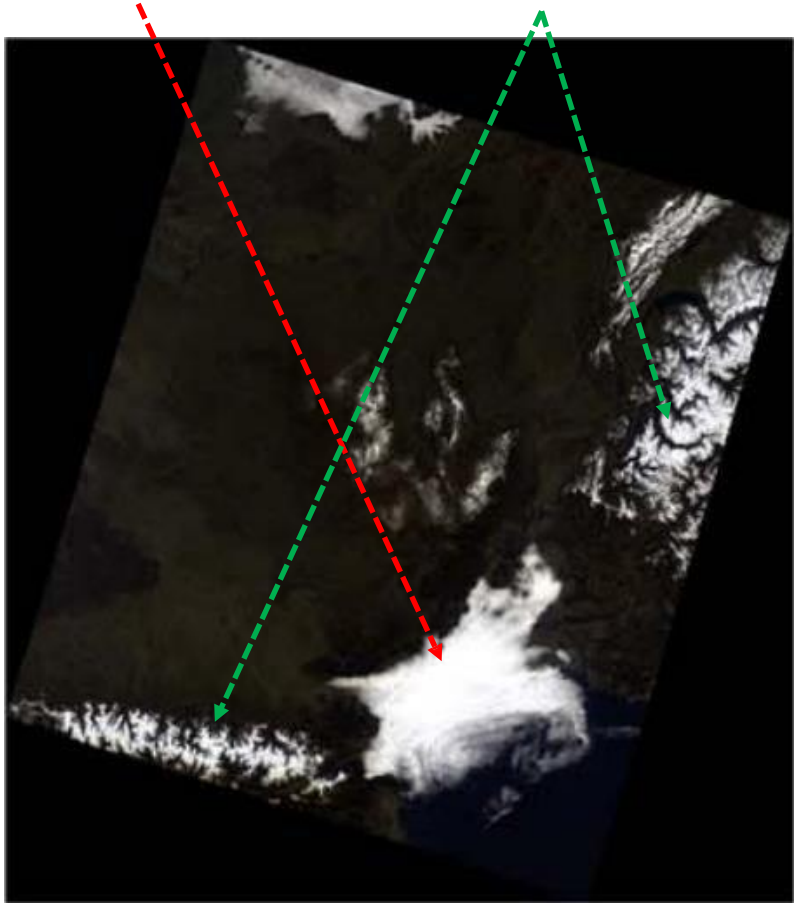
It is more complicated than that, but…



Shadows produced by clouds

SC One class-SVM (OC-SVM)

Clouds & No-clouds

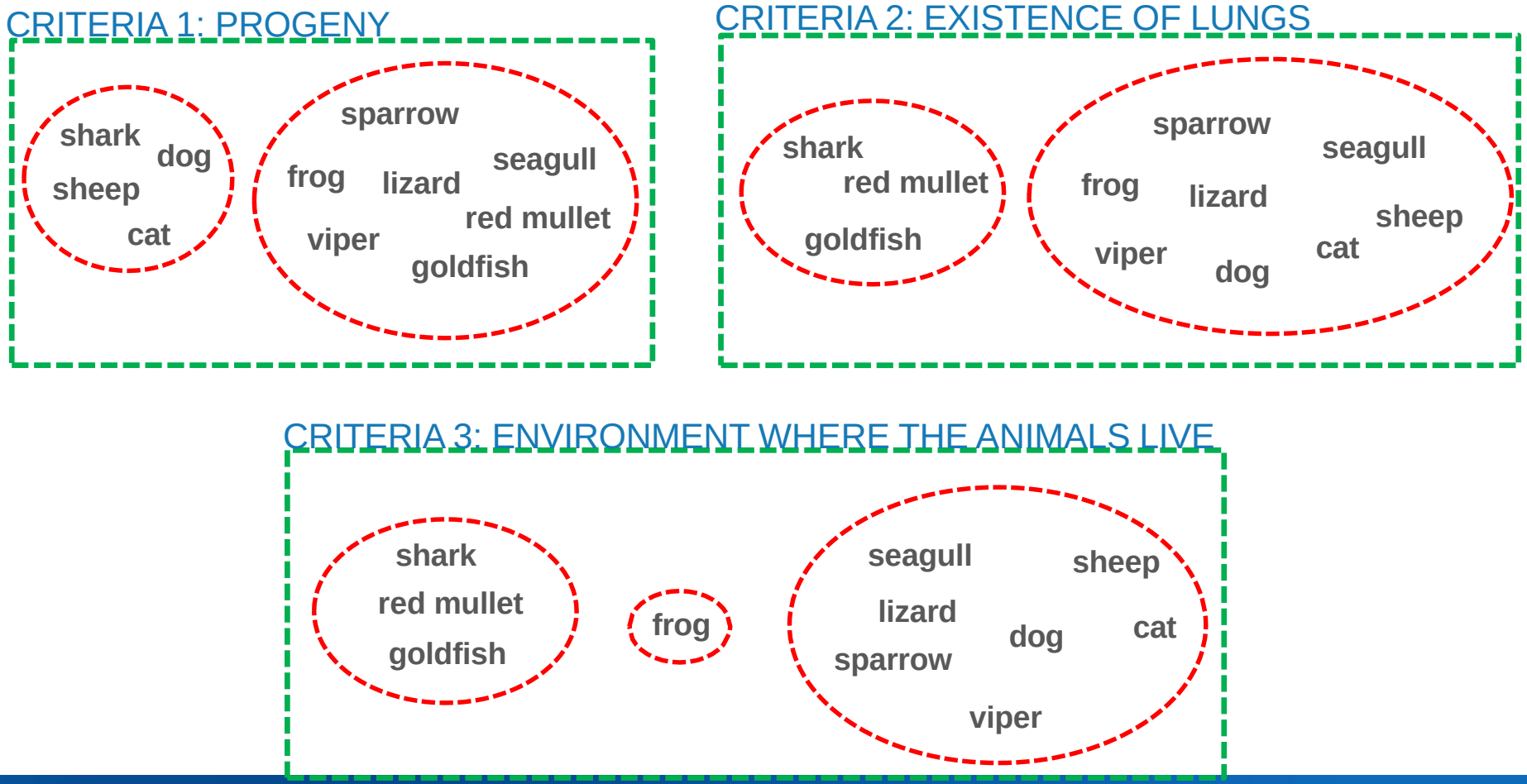


Unsupervised classification (UC)

Clustering

UC - Clustering

Imagine:



UC - Clustering

Clustering means “do clusters”.

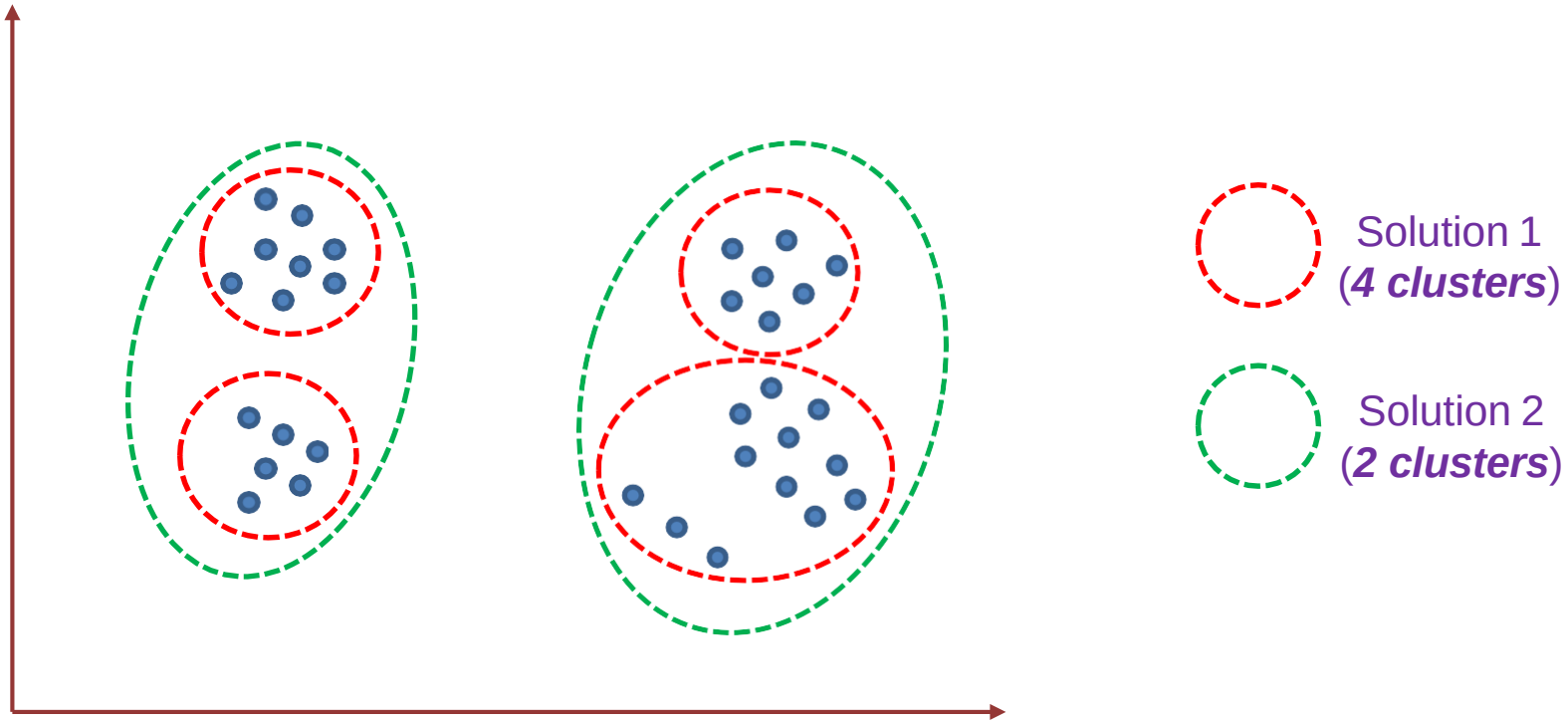
To do that:

- We need a **criterion**.
- Once the clusters are clustered we need a **validation** rule.
- Afterwards, we need to **interpret** the results.

UC - Clustering

Why do we need to validate the results?

Answer: *Because sometimes several solutions are possible.*



UC Clustering-Examples

K-Means

The idea is to minimise the distance from each data simple to a **representative** or **centroid**, i. e.:

Assume $\boldsymbol{S}$ sets: $\boldsymbol{S} = \{S_1, S_2, \dots, S_k\}$. The aim is to find the **partition** that accomplishes:

$$\arg\min_{S} \left\{ \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2 \right\}$$

UC Clustering-Examples

K-Means

This is accomplished iteratively, in two steps

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - \mu_i^{(t)} \right\|^2 \leq \left\| x_p - \mu_j^{(t)} \right\|^2 ; \quad \forall j, 1 \leq j \leq k \right\}$$

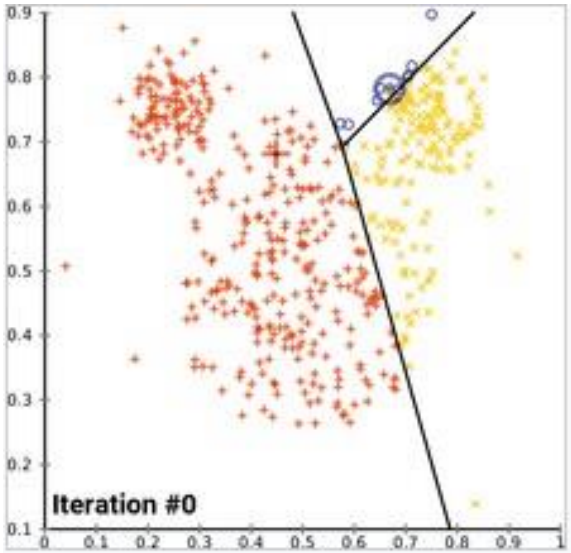
Assignment step:

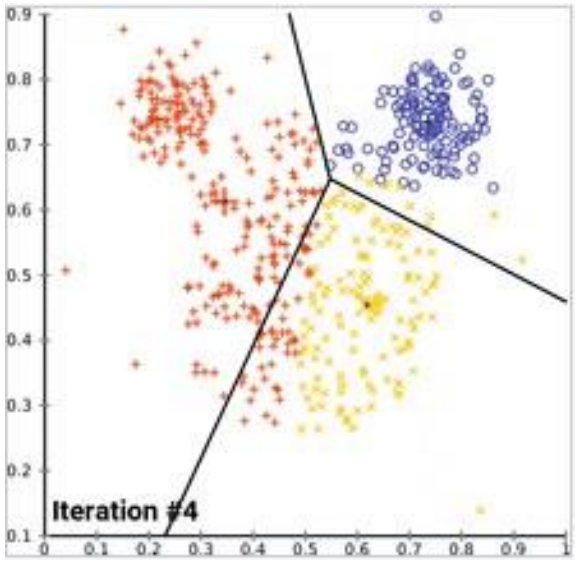
$$\mu_j^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

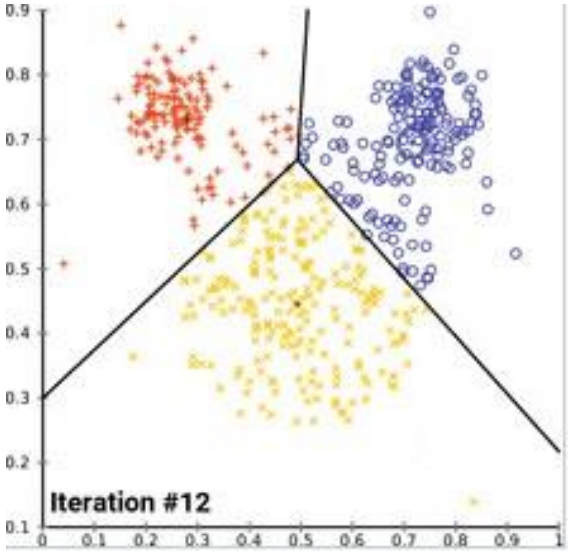
UC Clustering-Examples

K-Means

Memberships and centroids positions are iteratively updated

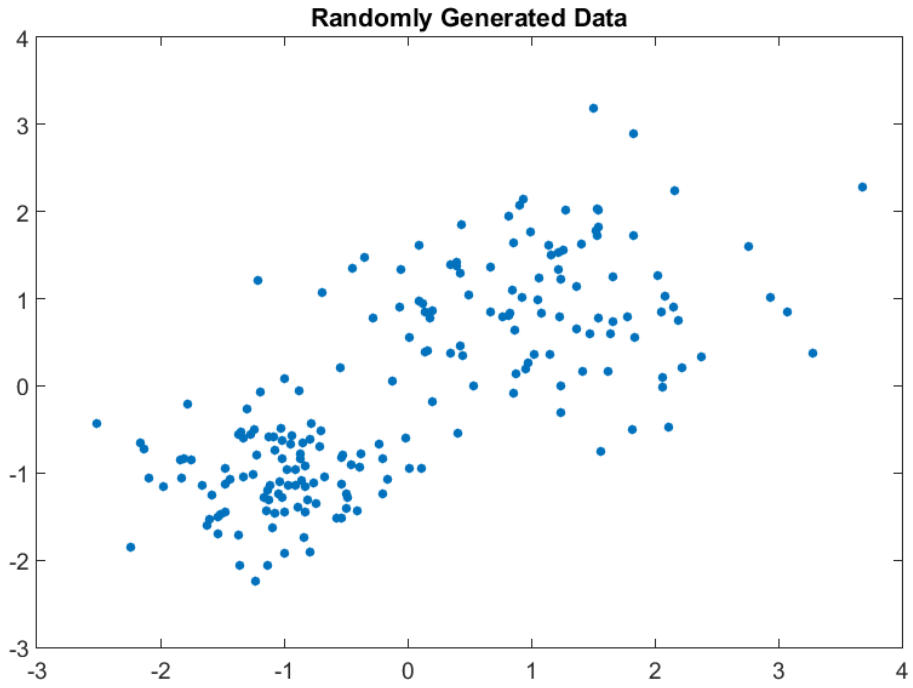


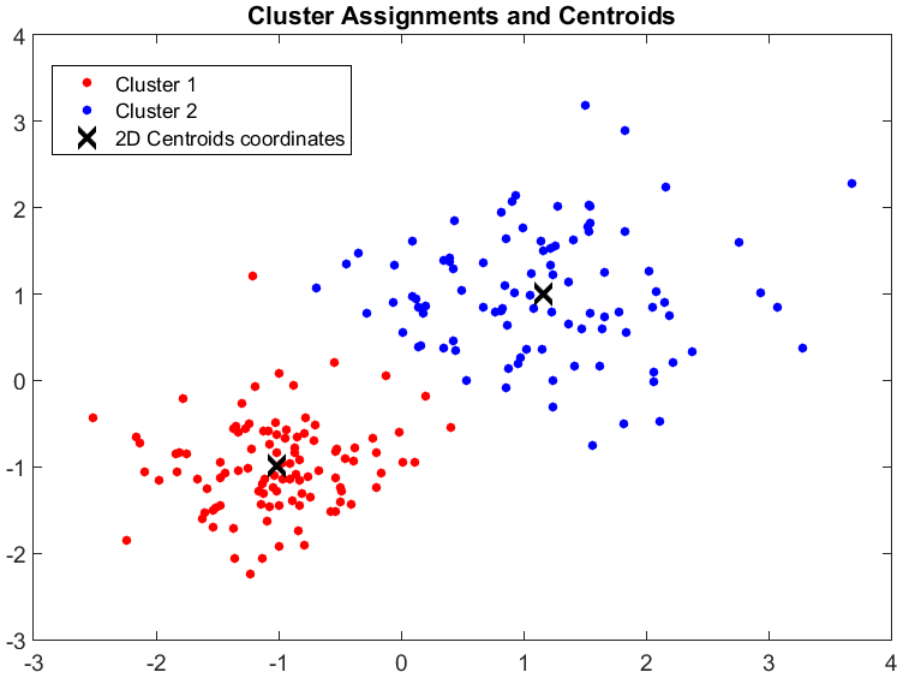




UC Clustering-Examples

K-Means

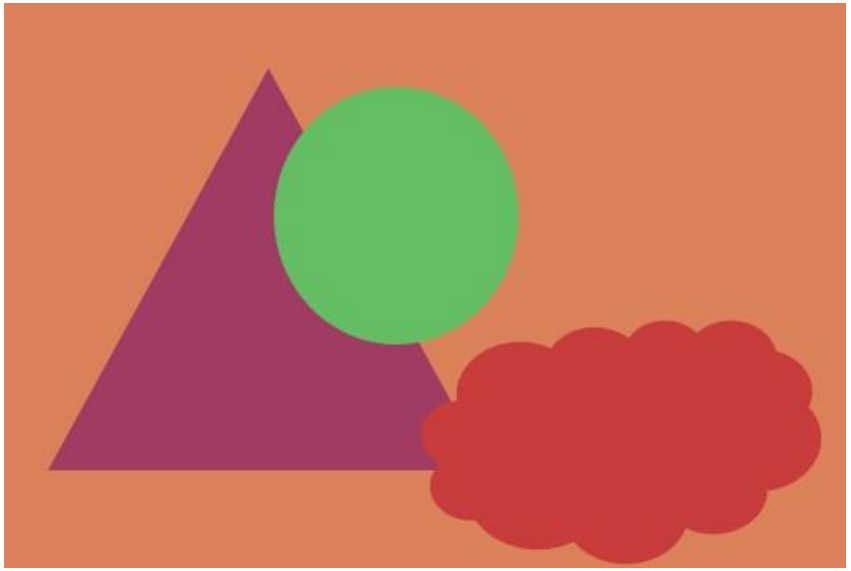




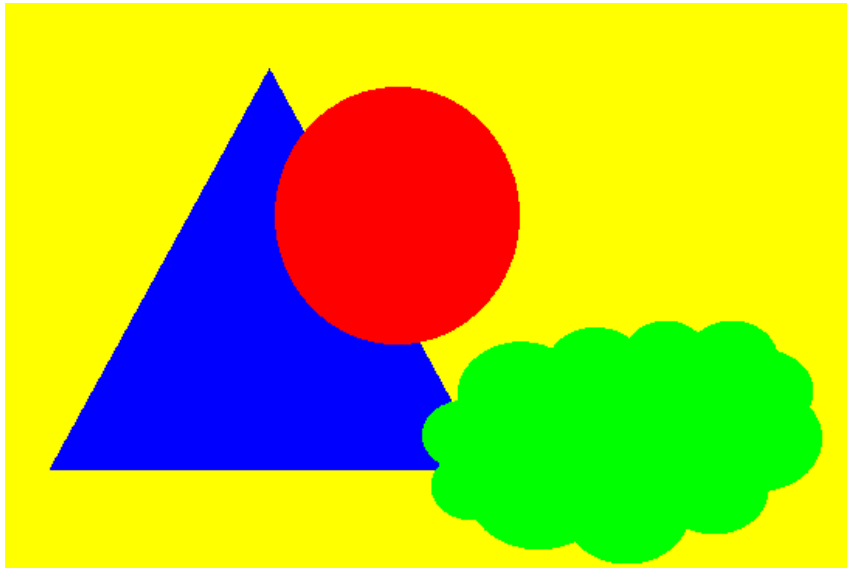
UC Clustering-Examples

K-Means – *Noise free*

Original – test image
4 clusters



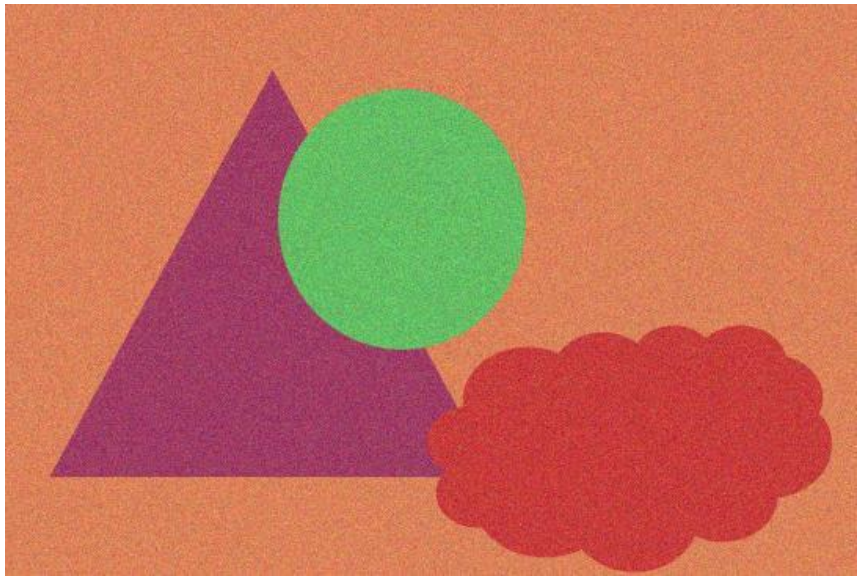
Segmentation / labeling result



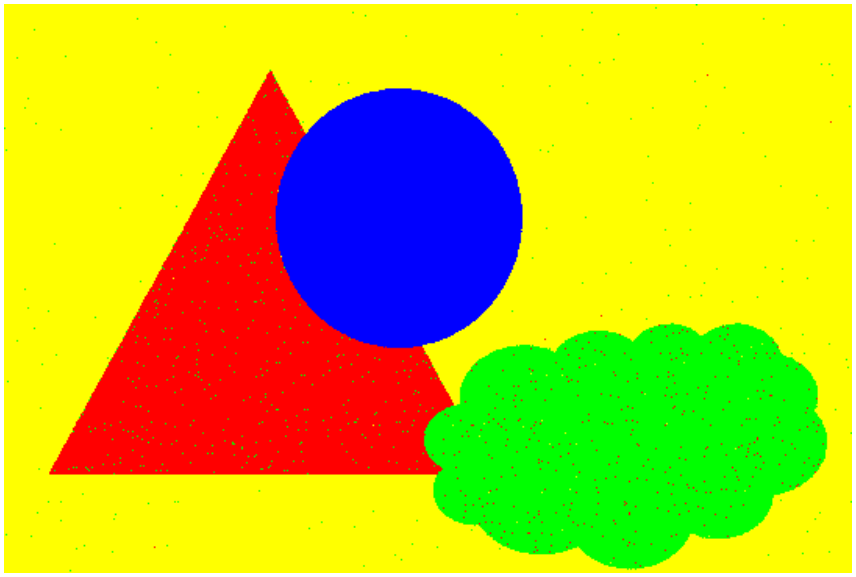
UC Clustering-Examples

K-Means – *Gaussian noise*

$\sigma = 0.05$



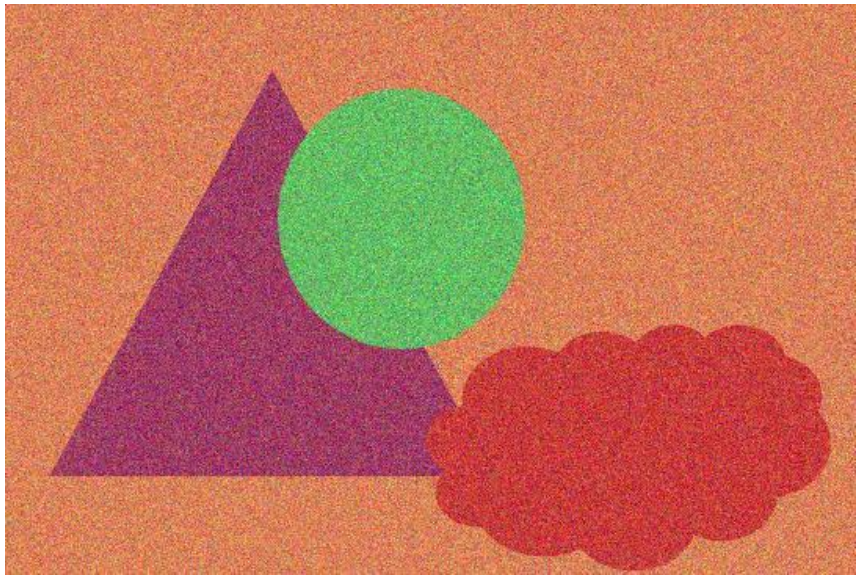
Segmentation / labeling result



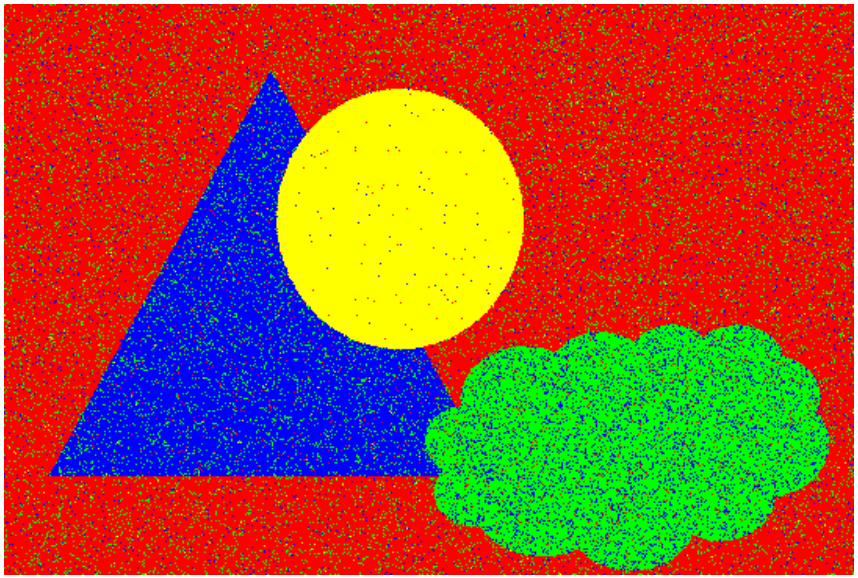
UC Clustering-Examples

K-Means – *Gaussian noise*

$\sigma = 0.1$



Segmentation / labeling result

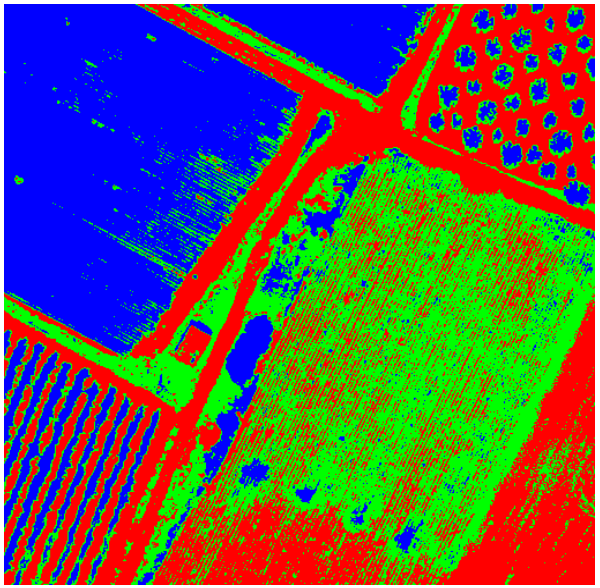


UC Clustering-Examples

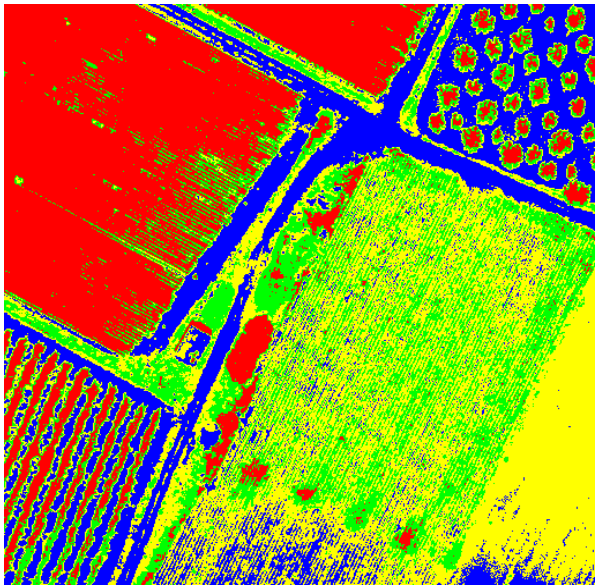
K-Means – *Examples from orthophotos*

Original RGB image

3 clusters



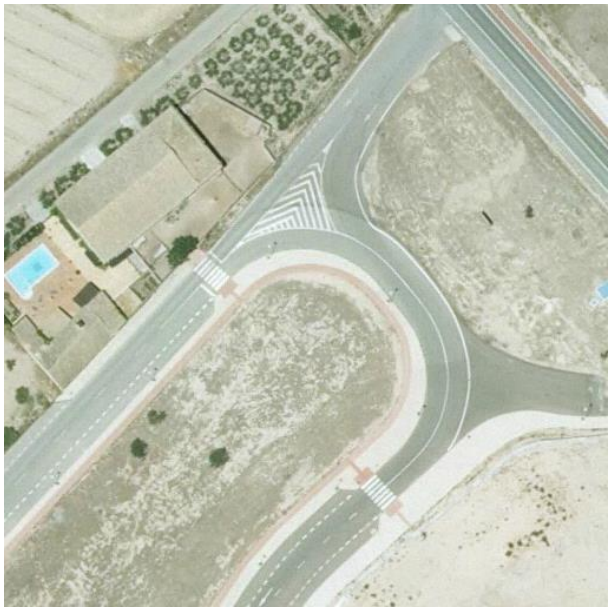
4 clusters



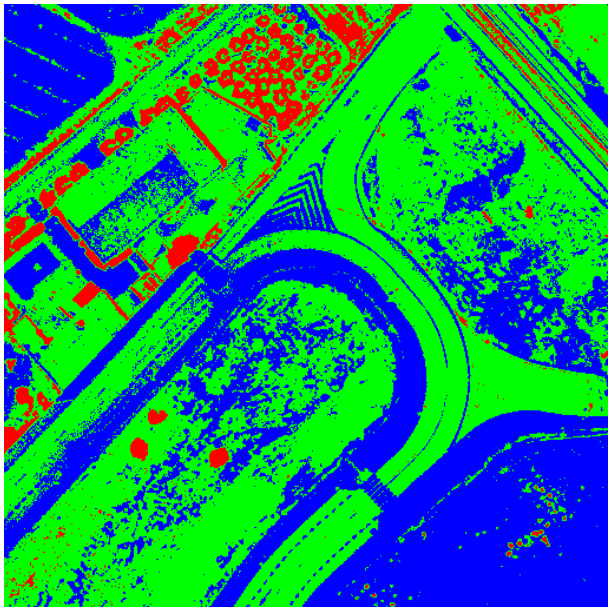
UC Clustering-Examples

K-Means – *Examples from orthophotos*

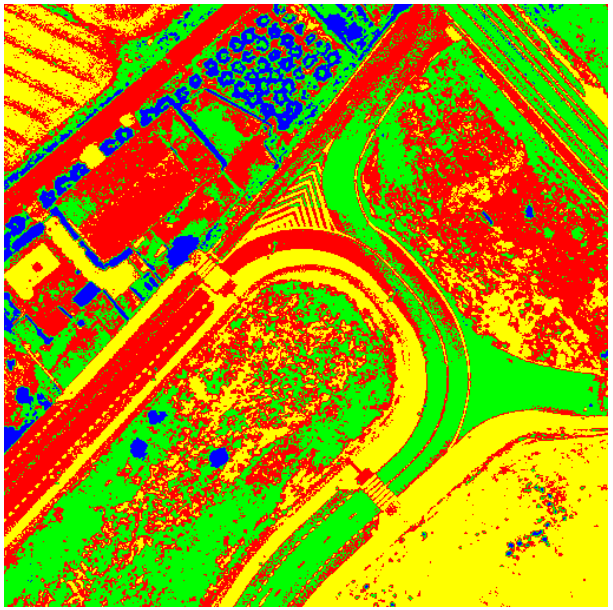
Original RGB image



3 clusters



4 clusters



Fuzzy C-Means (FCM)

In K-Means each data point belongs to a cluster with *probability* equal to 1.

In FCM the method allows each data point i to have a certain *membership* value to each cluster, j (represented by $\boldsymbol{u_{ij} \equiv u_i(x_j)}$).

The function to minimize would be:

$$\boldsymbol{J_m(U,v) = \sum_{j=1}^{n} \sum_{i=1}^{c} (u_{ij})^m \|x_j - \mu_i\|^2}$$

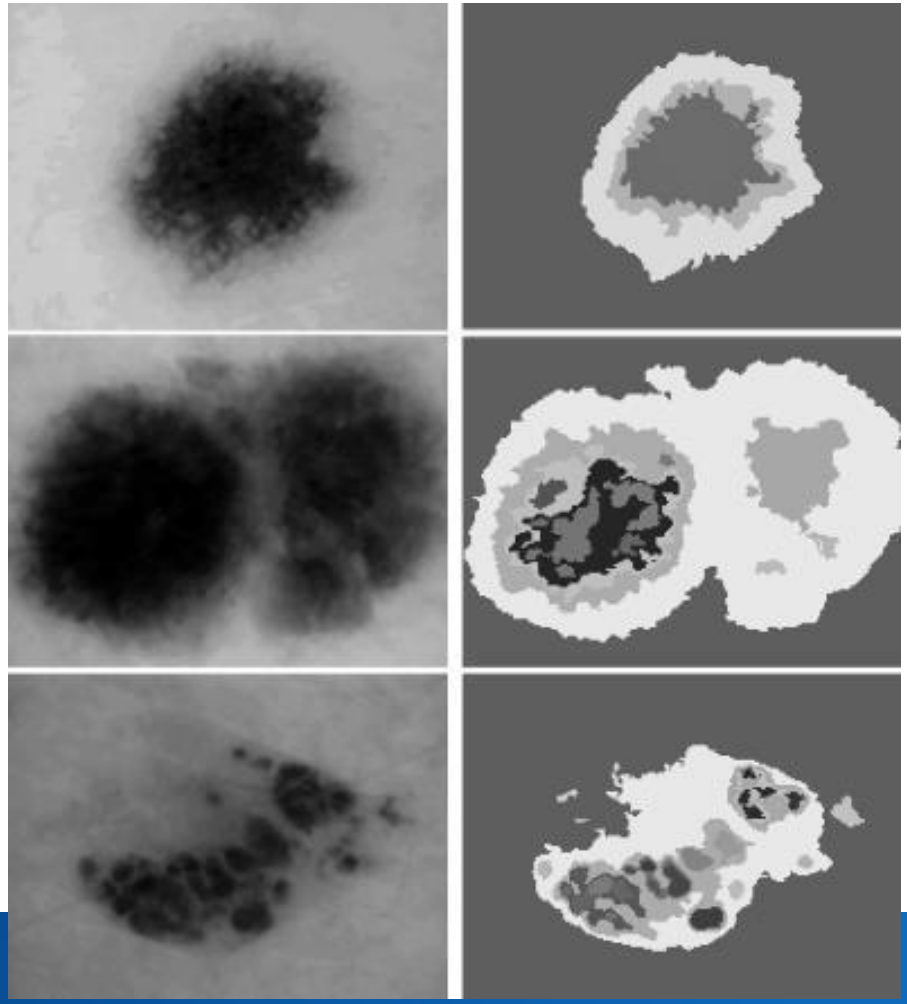
$$\boldsymbol{0 \le u_{ij} \le 1; \quad \sum_{i=1}^{c} u_{ij} = 1}$$

UC Clustering-Examples

Fuzzy C-Means (FCM)

Where could FCM work well?

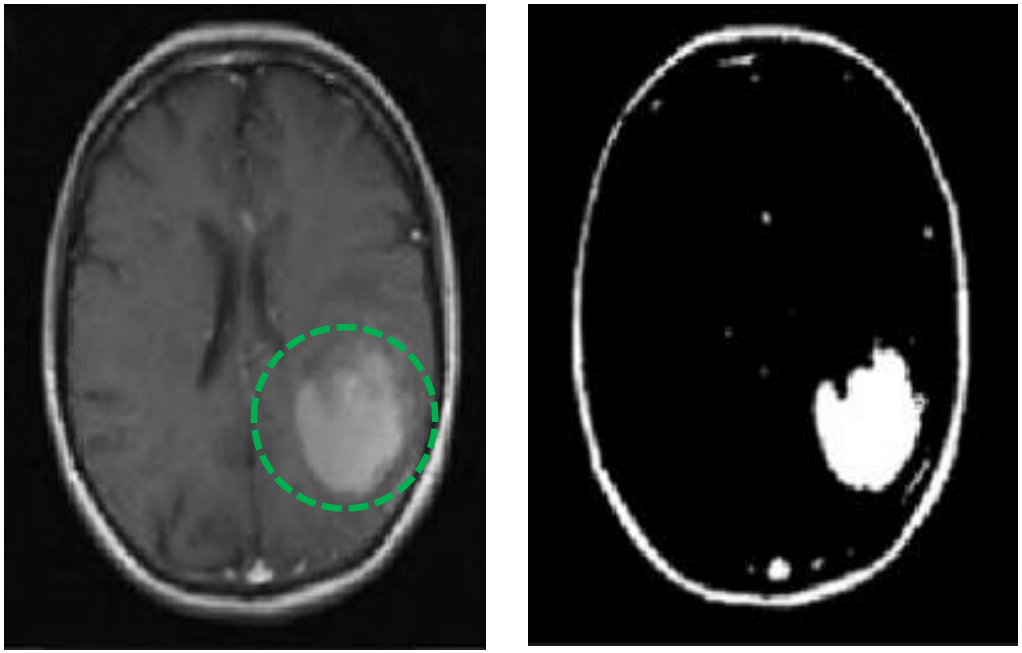
Answer: For instance, in objects with ***diffused borders***



UC Clustering-Examples

Fuzzy C-Means (FCM)

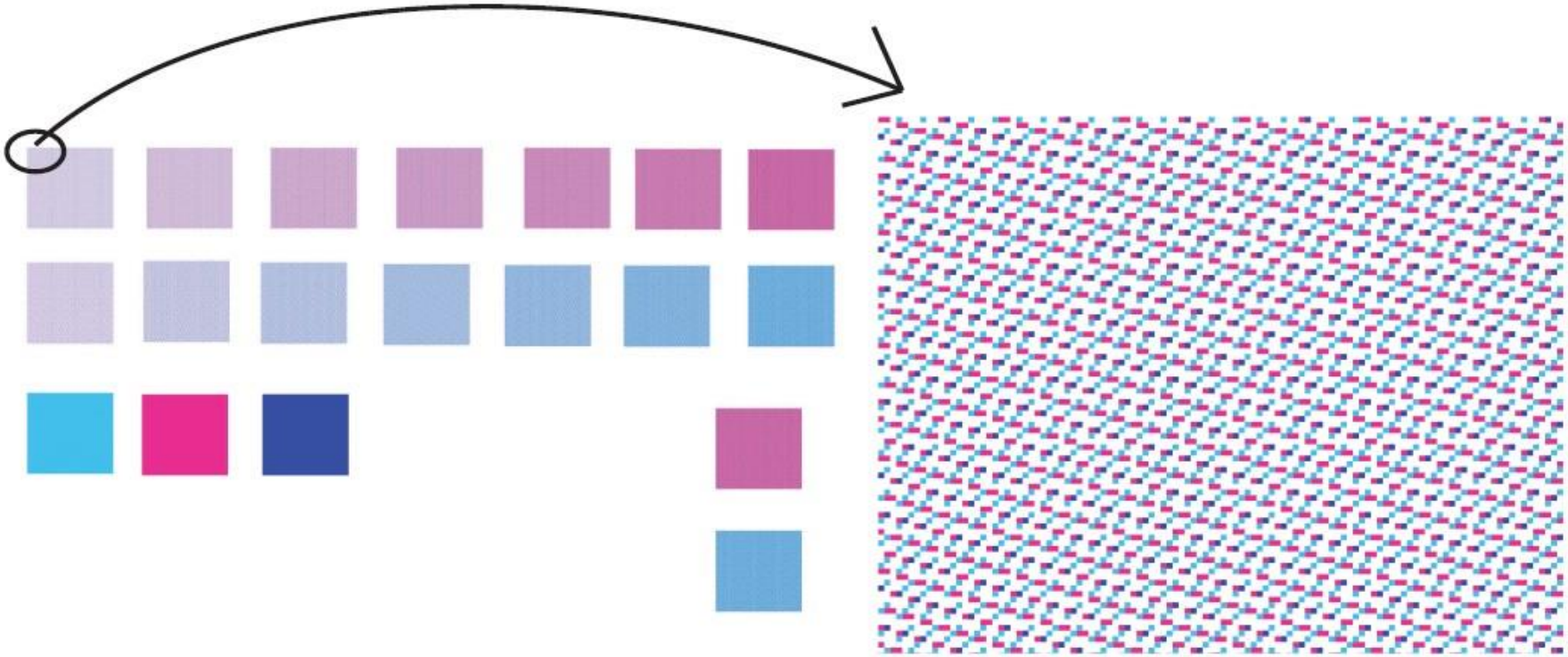
BRAIN



UC Clustering-Examples

Fuzzy C-Means (FCM)

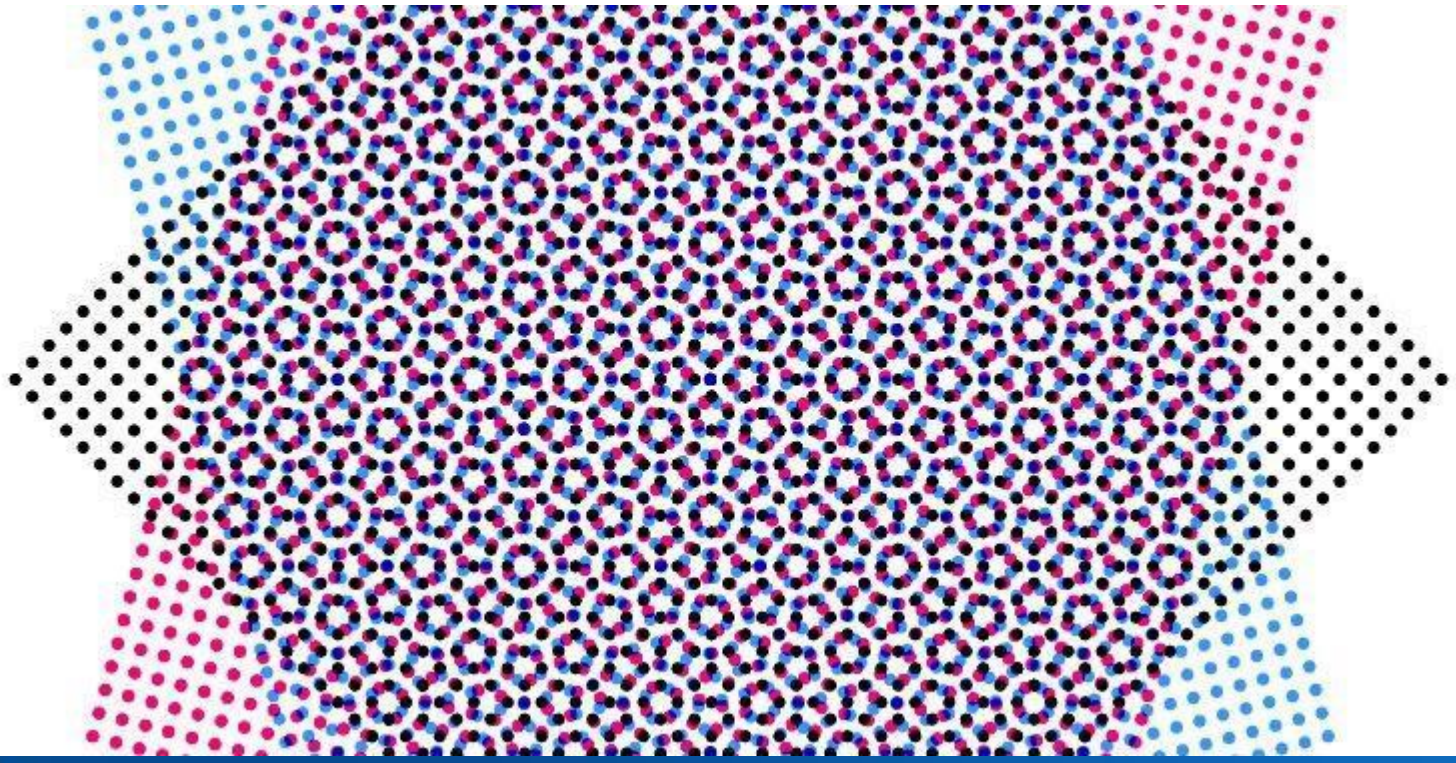
Digital file of a ***halftone*** (i. e., print/no print) CMYK design



UC Clustering-Examples

Fuzzy C-Means (FCM)

Be careful with the/a **MOIRÉ** pattern!



UC Clustering-Examples

Fuzzy C-Means (FCM)

A binary (CMYK) ink jet printer was used.

UC Clustering-Examples

Fuzzy C-Means (FCM)

UC Clustering-Examples

Fuzzy C-Means (FCM)

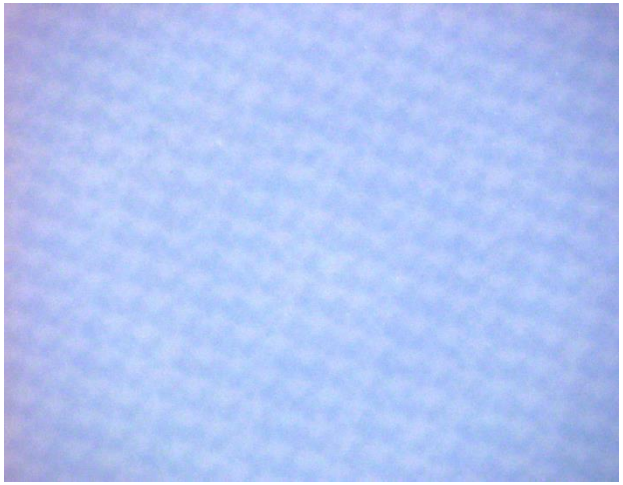
Images of the printed ceramic tile at microscopic scale were acquired:

UC Clustering-Examples

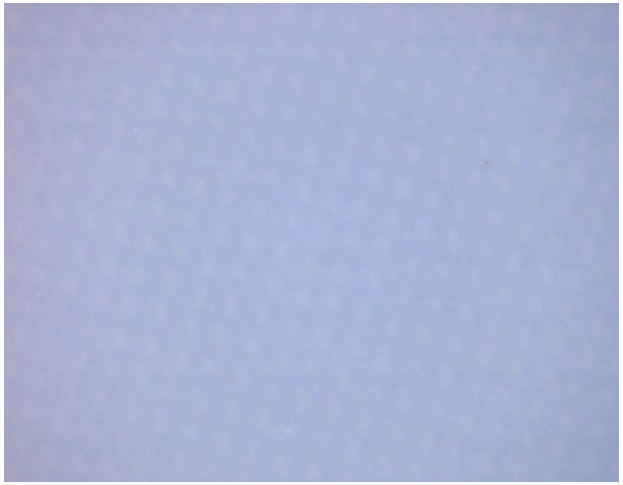
Fuzzy C-Means (FCM)

$C = 20\%$

$C = 50\%$



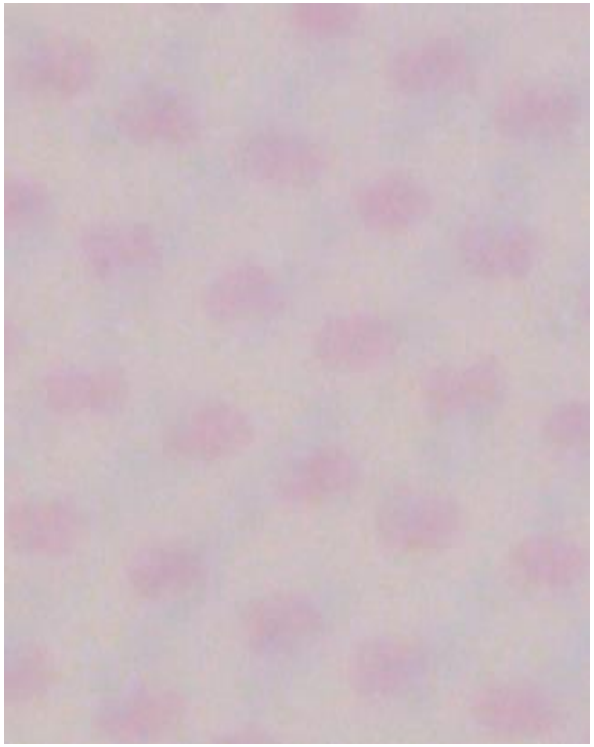
$C = 80\%$



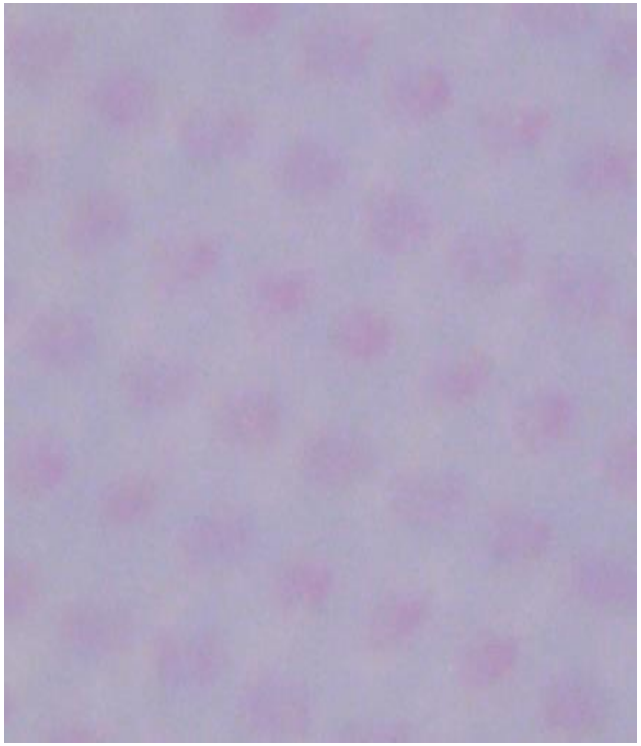
UC Clustering-Examples

Fuzzy C-Means (FCM)

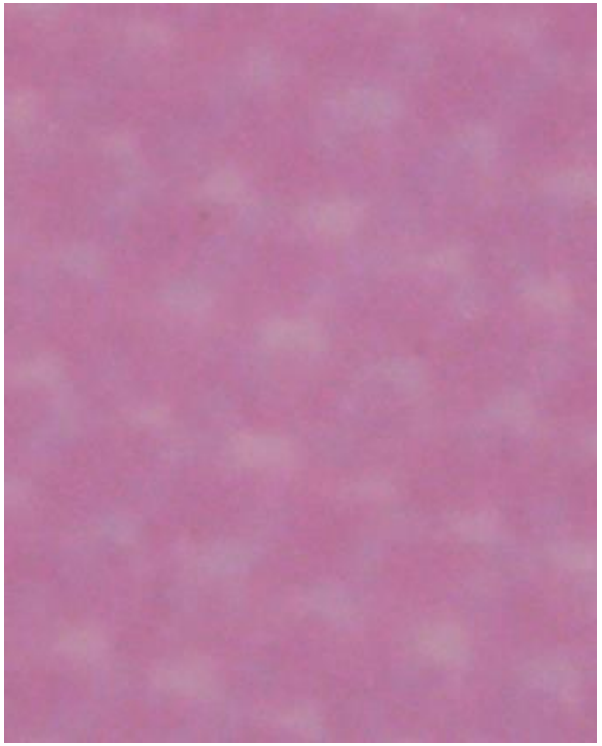
$C = 20\%$; $M = 20\%$



$C = 80\%$; $M = 20\%$



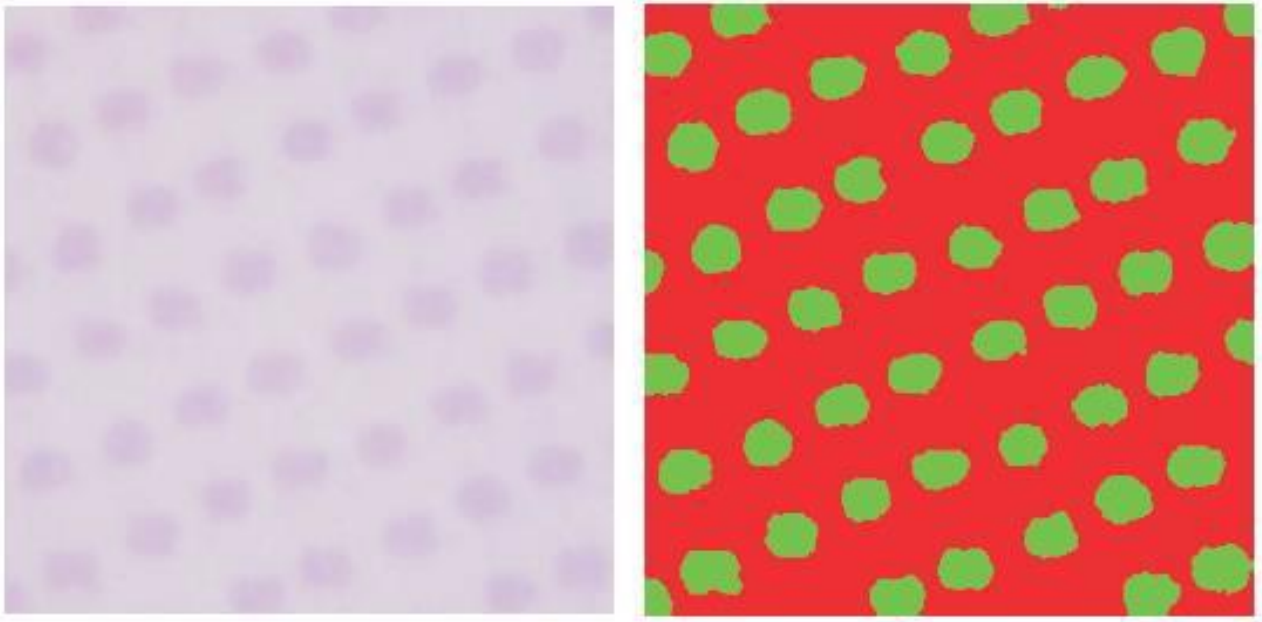
$C = 20\%$; $M = 80\%$



UC Clustering-Examples

Fuzzy C-Means (FCM)

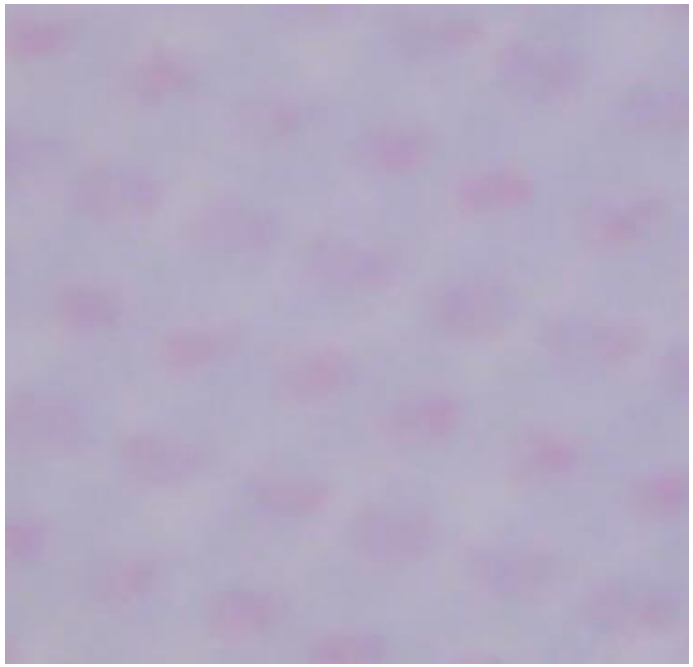
Microscopic image of *Magenta* (color) printed dots on a ceramic tile

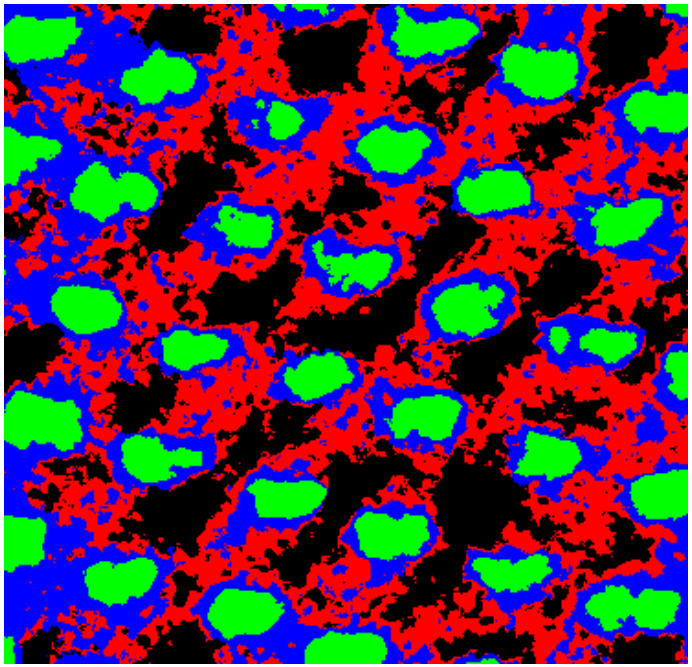


UC Clustering-Examples

Fuzzy C-Means (FCM)

$$\boldsymbol{C = 70\%; M = 20\%}$$





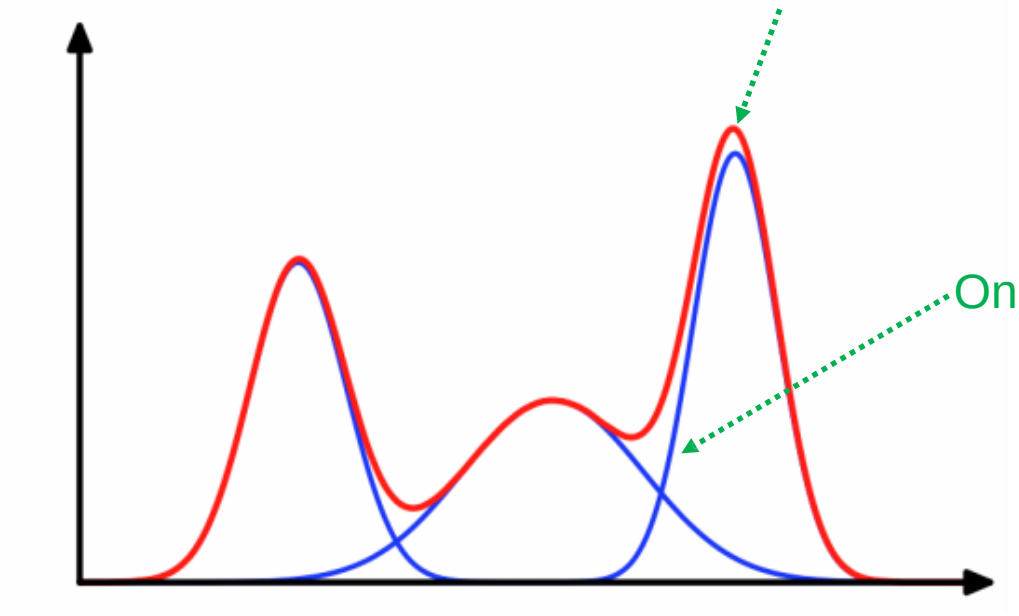
UC Clustering-Examples

Mixtures of Gaussians – Expectation Maximization (EM)

What is a **Mixture of Gaussians (*MoGs*)**?

$$p(x) = \sum_{j=1}^{M} p(\Omega_j) \cdot p(x|\Omega_j)$$

Evolving curve –
Linear combination of Gaussians

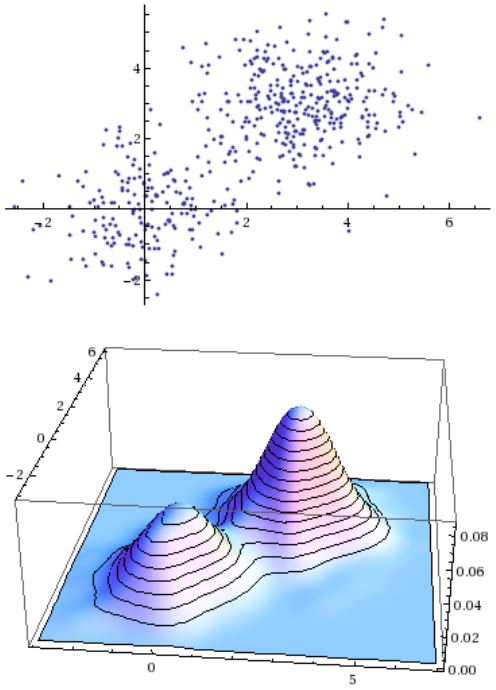


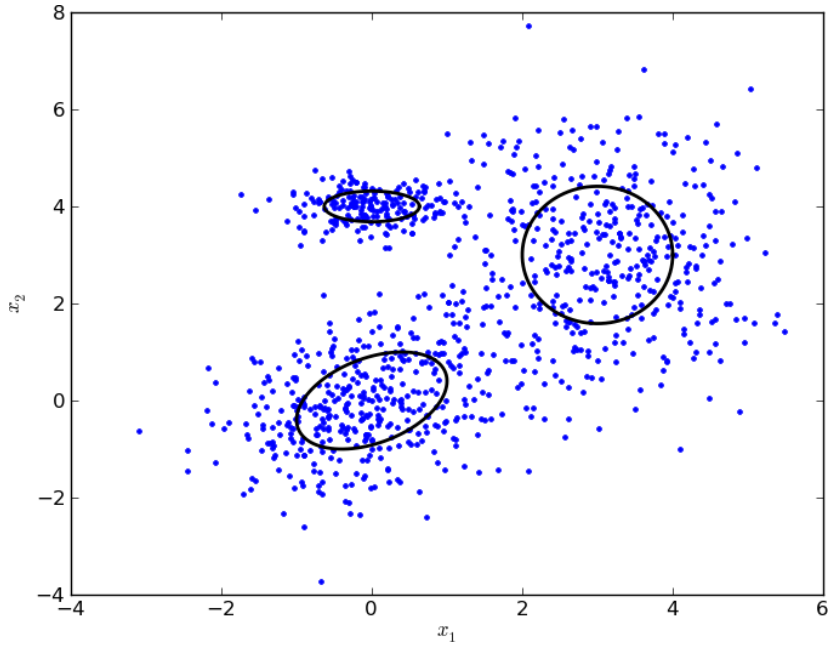
One 1D Gaussian

UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

What would we have in 2D?





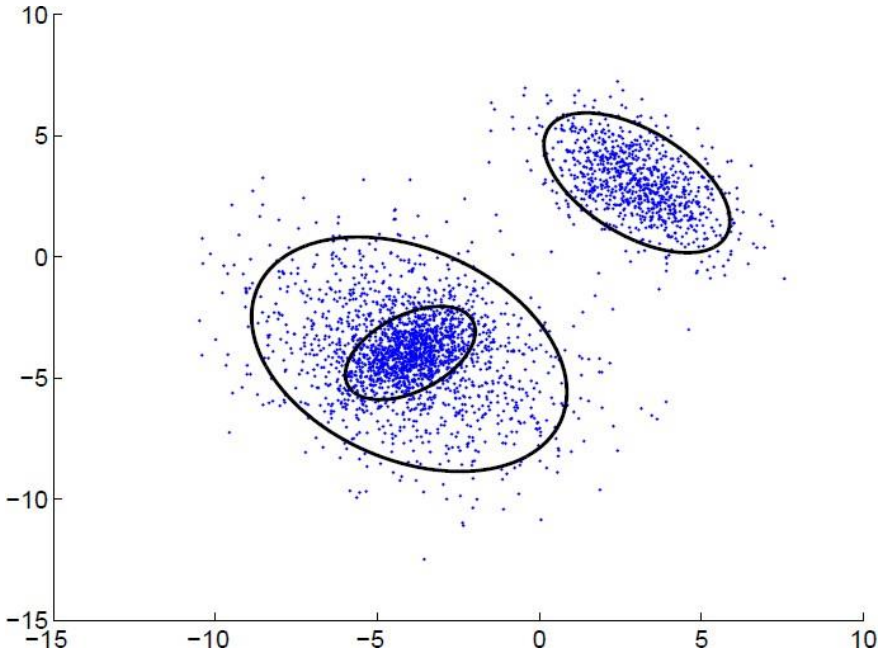
UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

In a mixture of gaussians, we have:

$$p(x) = \sum_{j=1}^{M} p(\Omega_j) \cdot p(x|\Omega_j)$$

$$p(x|\Omega_j) = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} e^{\left[-\frac{1}{2} \cdot \left((x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\right)\right]}$$



UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

From Bayes rule and

$$p(x) = \sum_{j=1}^{M} p(\Omega_j) \cdot p(x|\Omega_j); \qquad p(x|\Omega_j) = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} e^{\left[-\frac{1}{2} \cdot \left((x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\right)\right]}$$

we have:

$$\boldsymbol{p(\Omega_j|x) = \frac{p(\Omega_j) \cdot p(x|\Omega_j)}{\sum_{k=1}^{M} p(\Omega_k) \cdot p(x|\Omega_k)}}$$

UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

Expectation (E) step:

$$p(\Omega_j|x_i)^{(t)} = \frac{p(\Omega_j)^{(t)} \cdot p(x_i|\Omega_j)^{(t)}}{\sum_k p(\Omega_k)^{(t)} \cdot p(x_i|\Omega_k)^{(t)}}$$

Maximization (M) step:

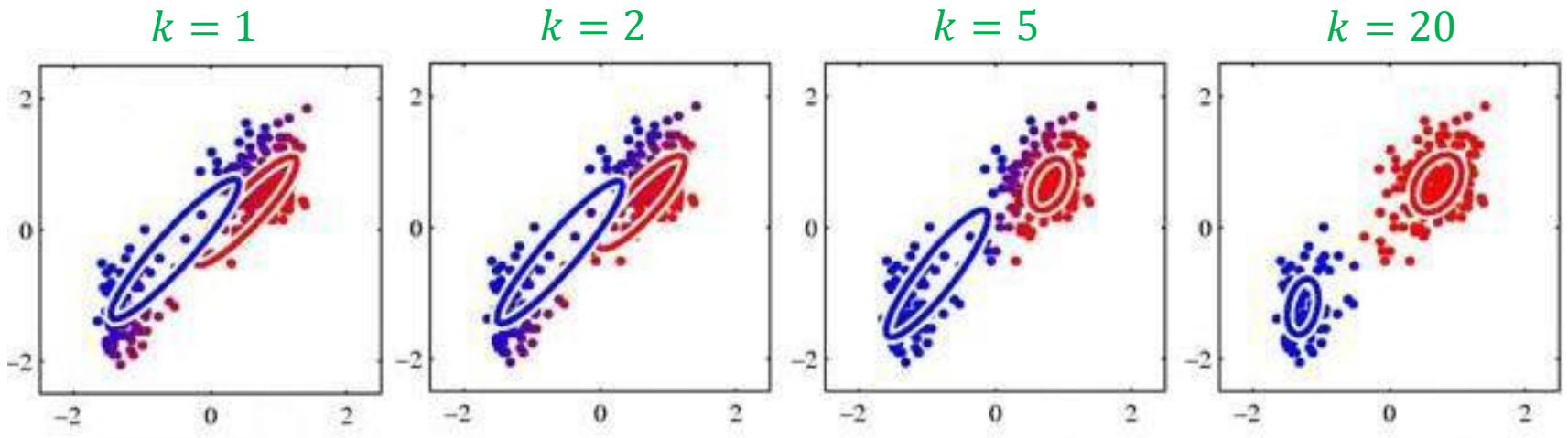
$$p(\Omega_j)^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \left\{ p(\Omega_j|x_i)^{(t)} \right\}$$

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} \left\{ x_i p(\Omega_j|x_i)^{(t)} \right\}}{\sum_{i=1}^{n} \left\{ p(\Omega_j|x_i)^{(t)} \right\}}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{n} \left\{ p(\Omega_j|x_i)^{(t)} \right\} (x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^T}{\sum_{i=1}^{n} \left\{ p(\Omega_j|x_i)^{(t)} \right\}}$$

UC Clustering-Examples

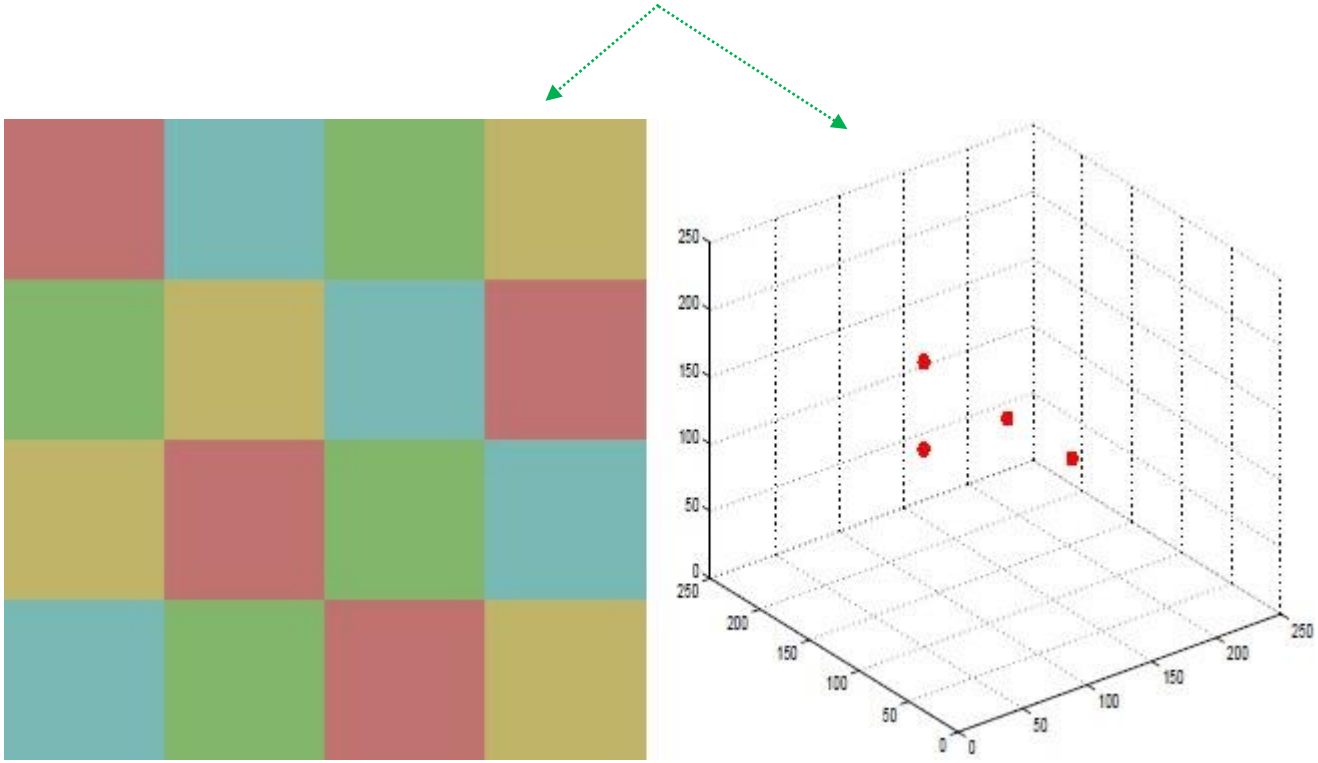
Mixtures of Gaussians– Expectation Maximization (EM)



UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

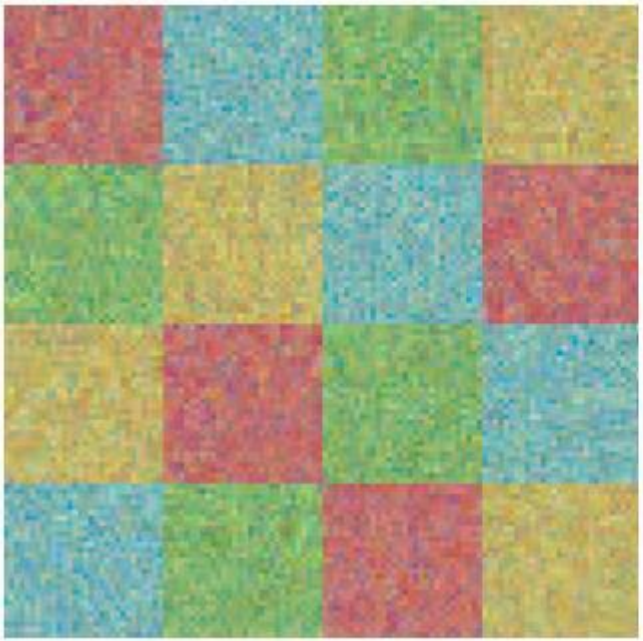
We have a “clean image”

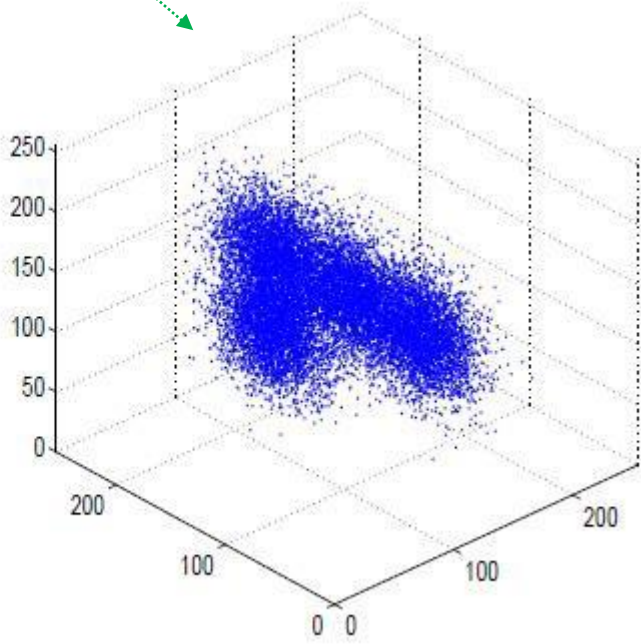


UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

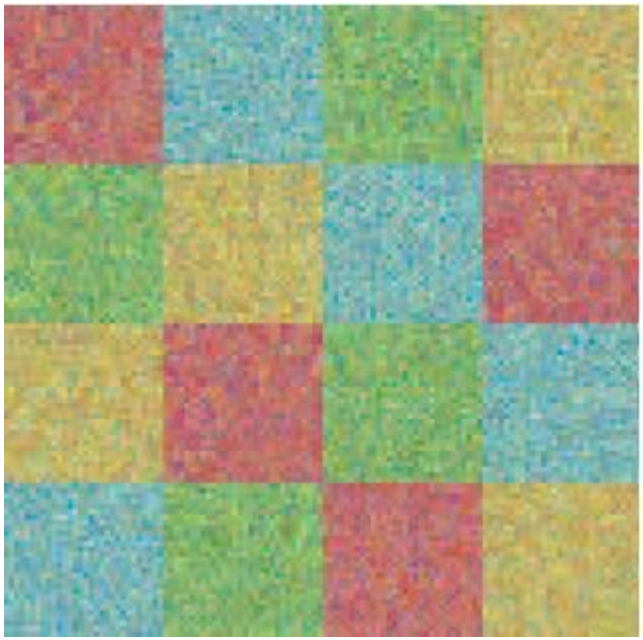
We introduce “gaussian noise”

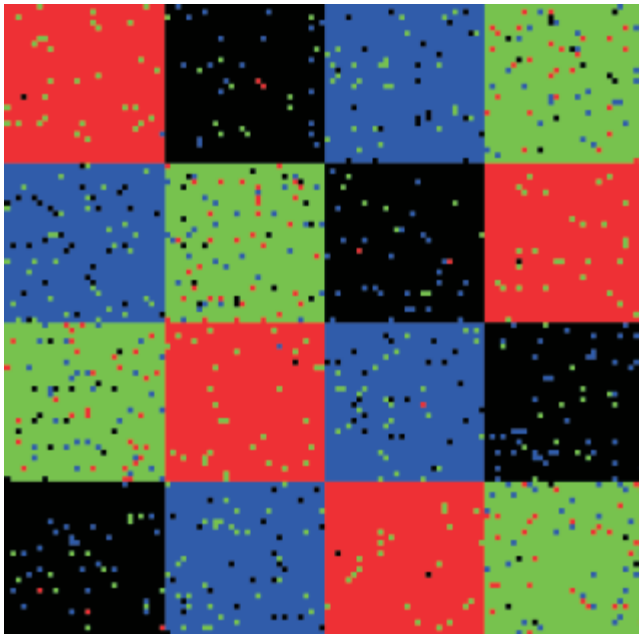




UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

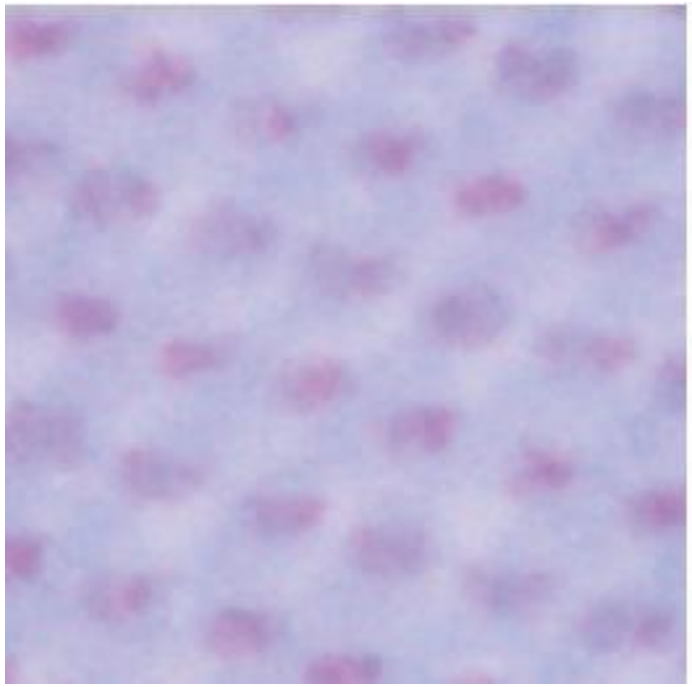


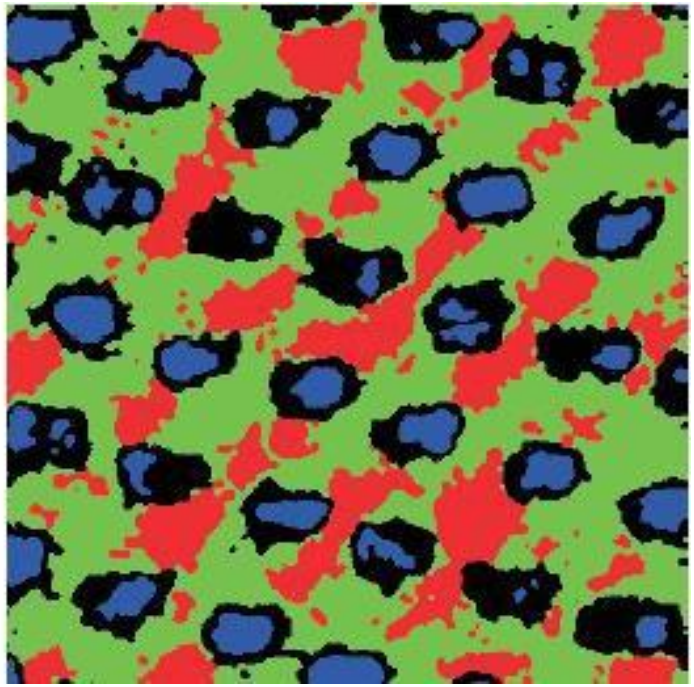


UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

$\boldsymbol{C = 80\%;} \quad \boldsymbol{M = 20\%}$

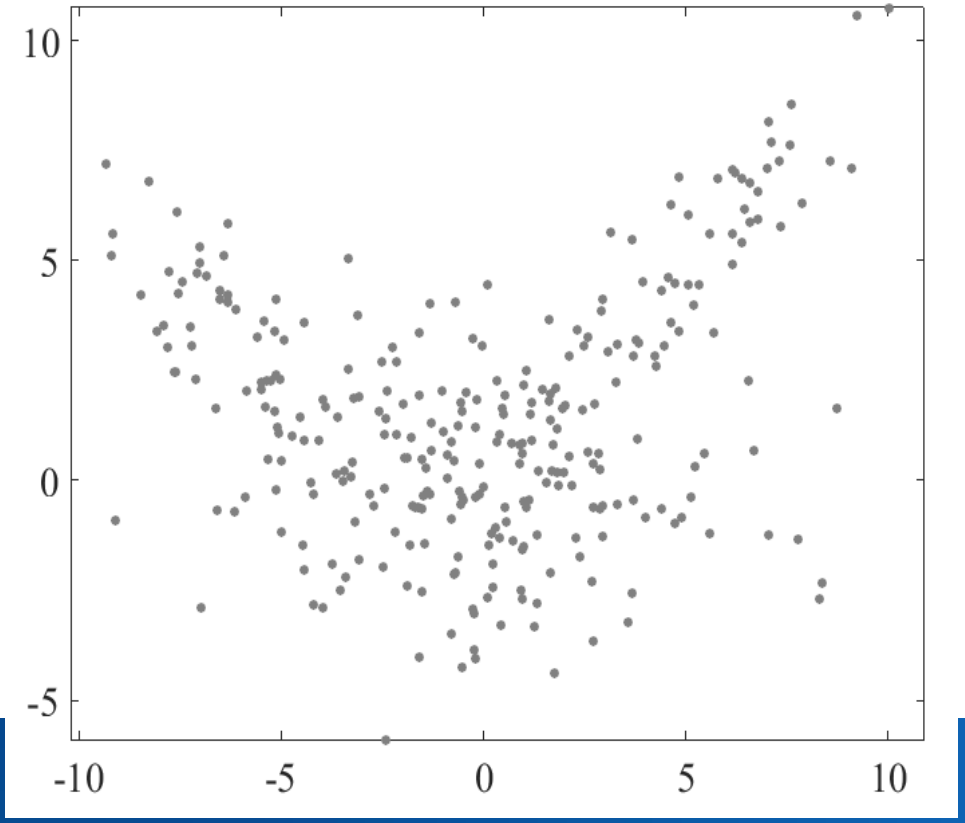




UC Clustering-Examples

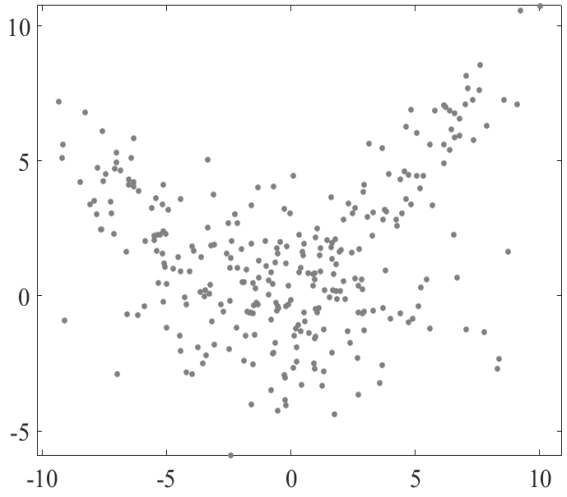
Mixtures of Gaussians– Expectation Maximization (EM)

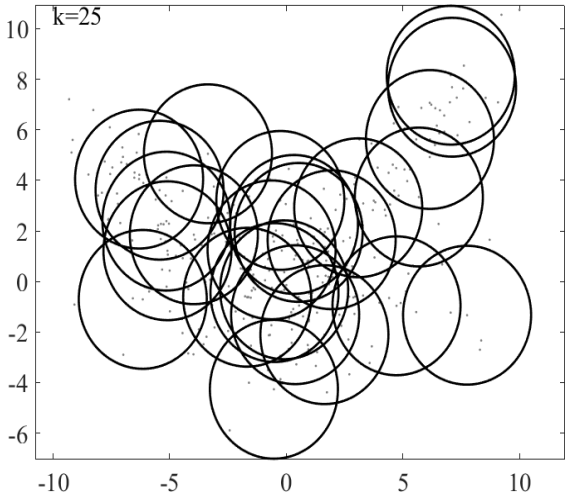
What if we can *tune* or *select* or *infer* the number of gaussians?

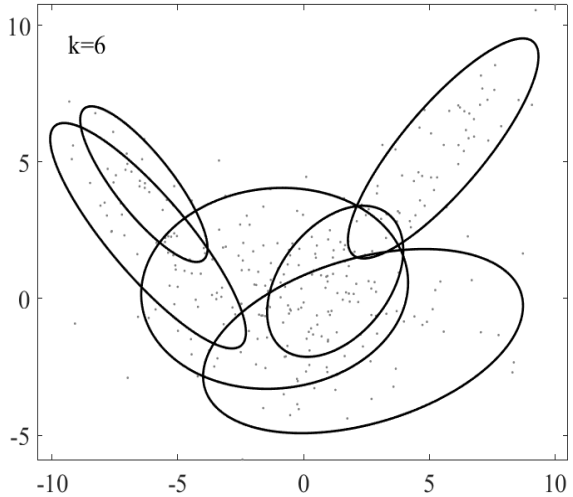


UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

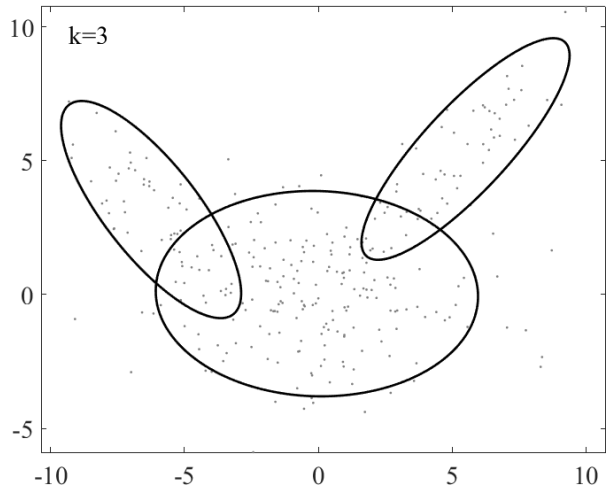


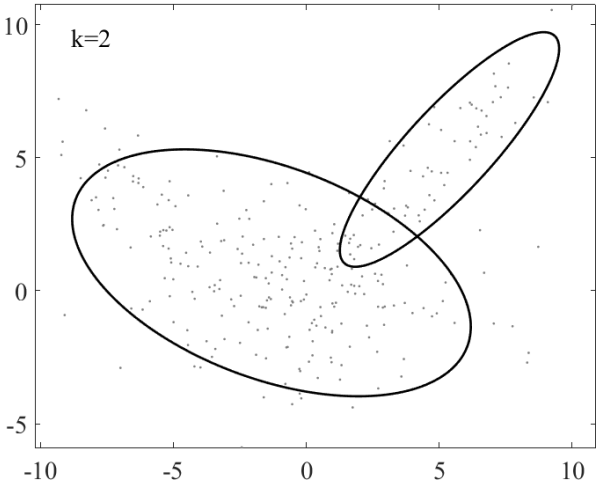


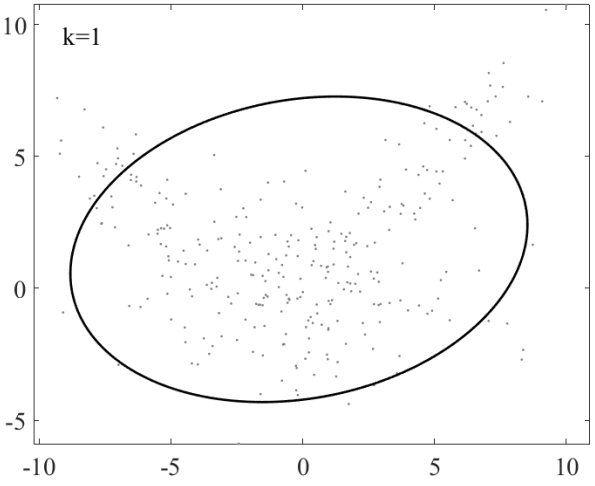


UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

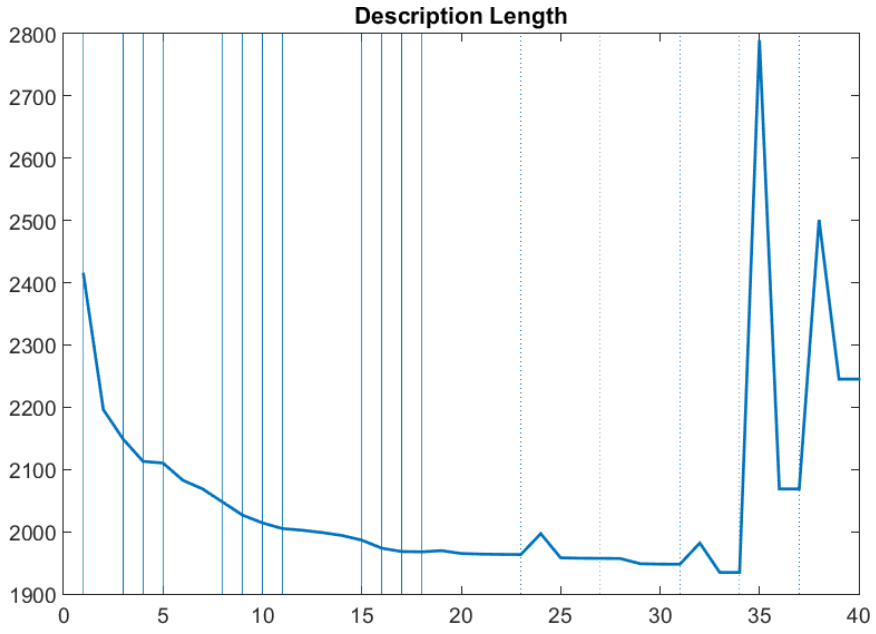


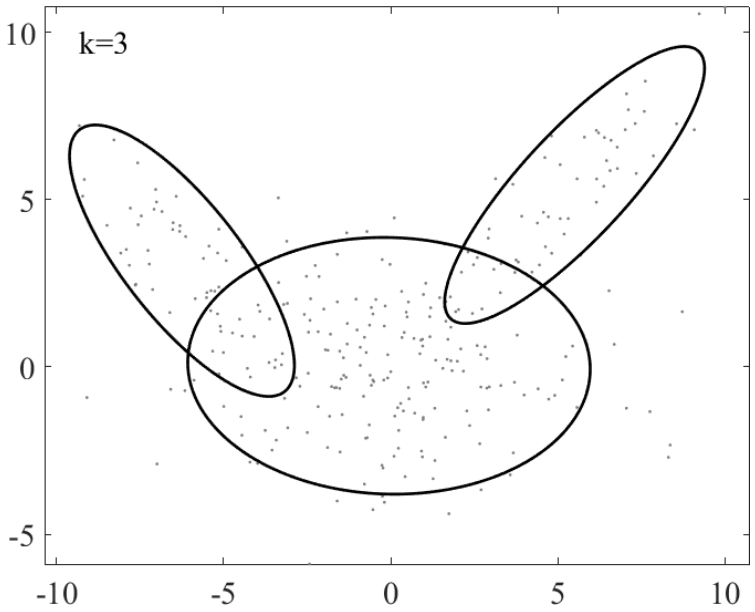




UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

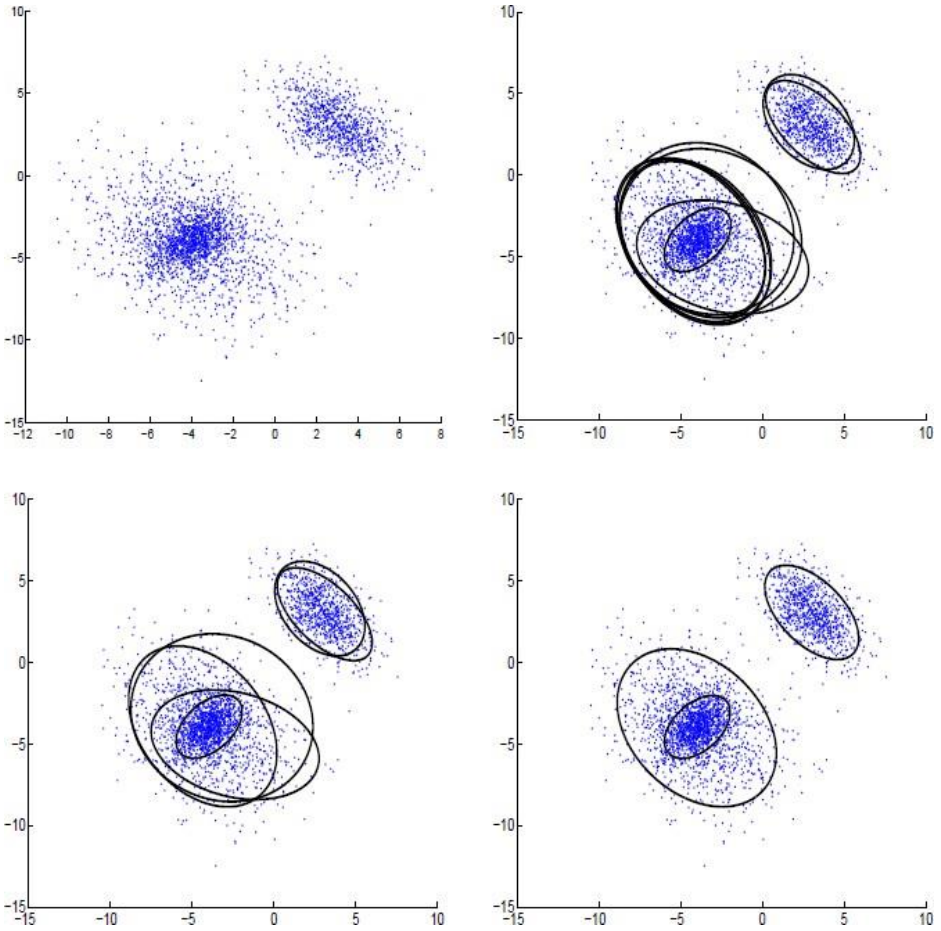




UC Clustering-Examples

Mixtures of Gaussians– Expectation Maximization (EM)

No *a priori* knowledge about the data!

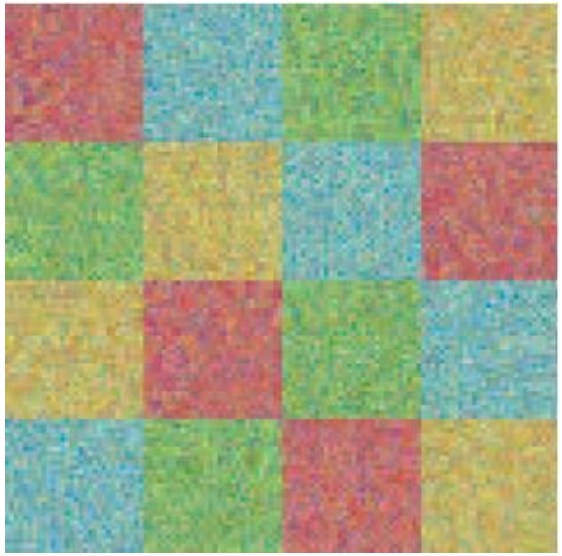


UC Clustering-Examples

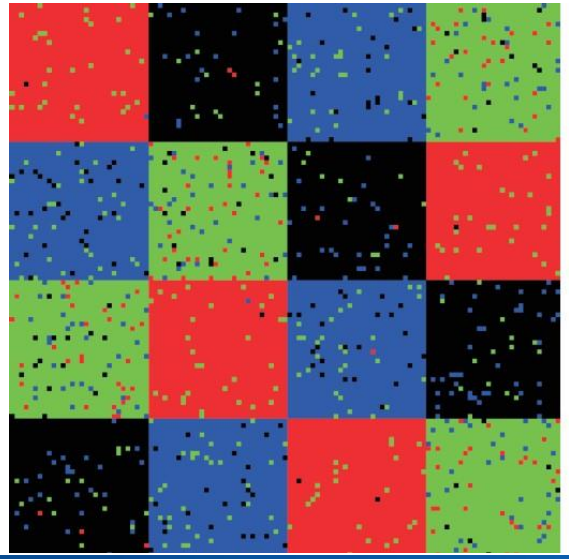
Can we improve the segmentation results?

YES! If we include information from the neighbourhood.

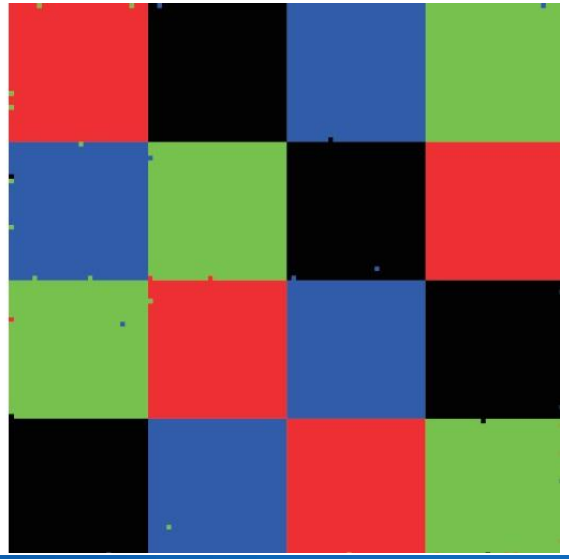
Original



EM



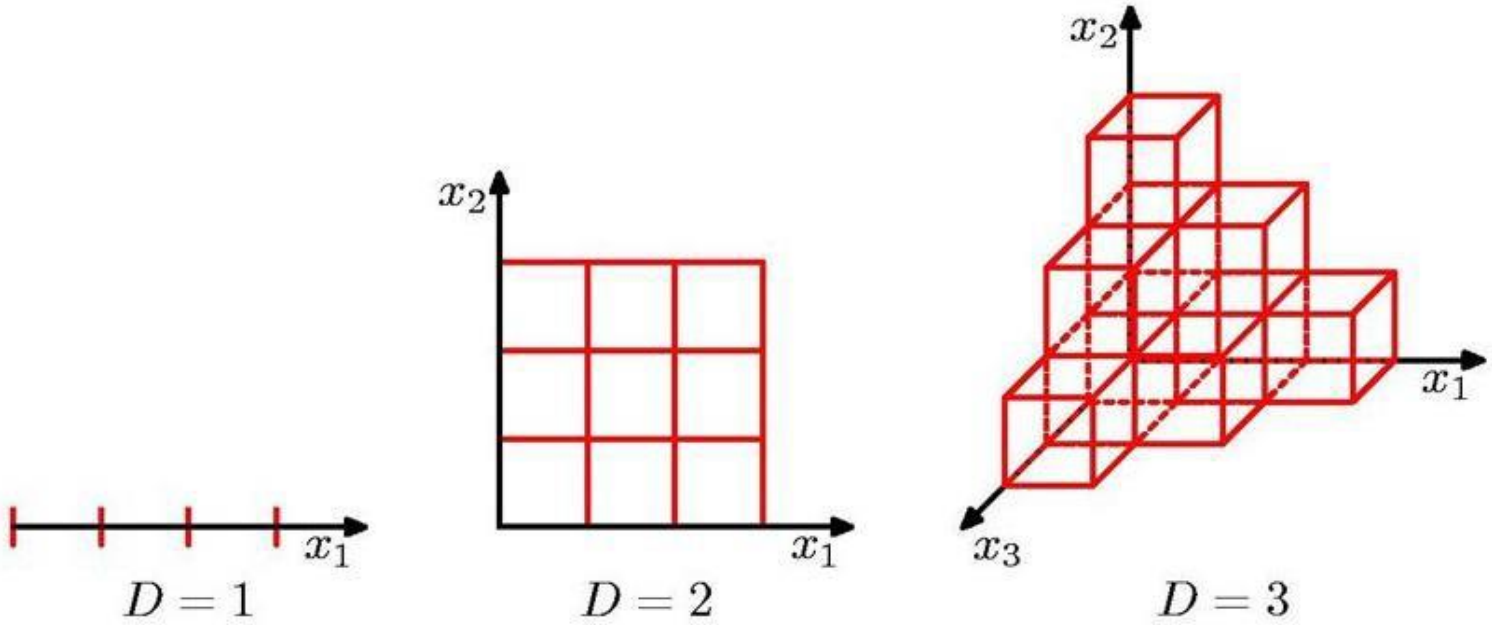
Iterative Conditional Modes (ICM)



Houston, we have a few problems!

Problems

1) *Curse* of dimensionality (I)



Problems

Curse of dimensionality

The Golden rule would be:

$$n \approx 10^d$$

So that the **number** of points (n) needed for a reliable training (and testing):

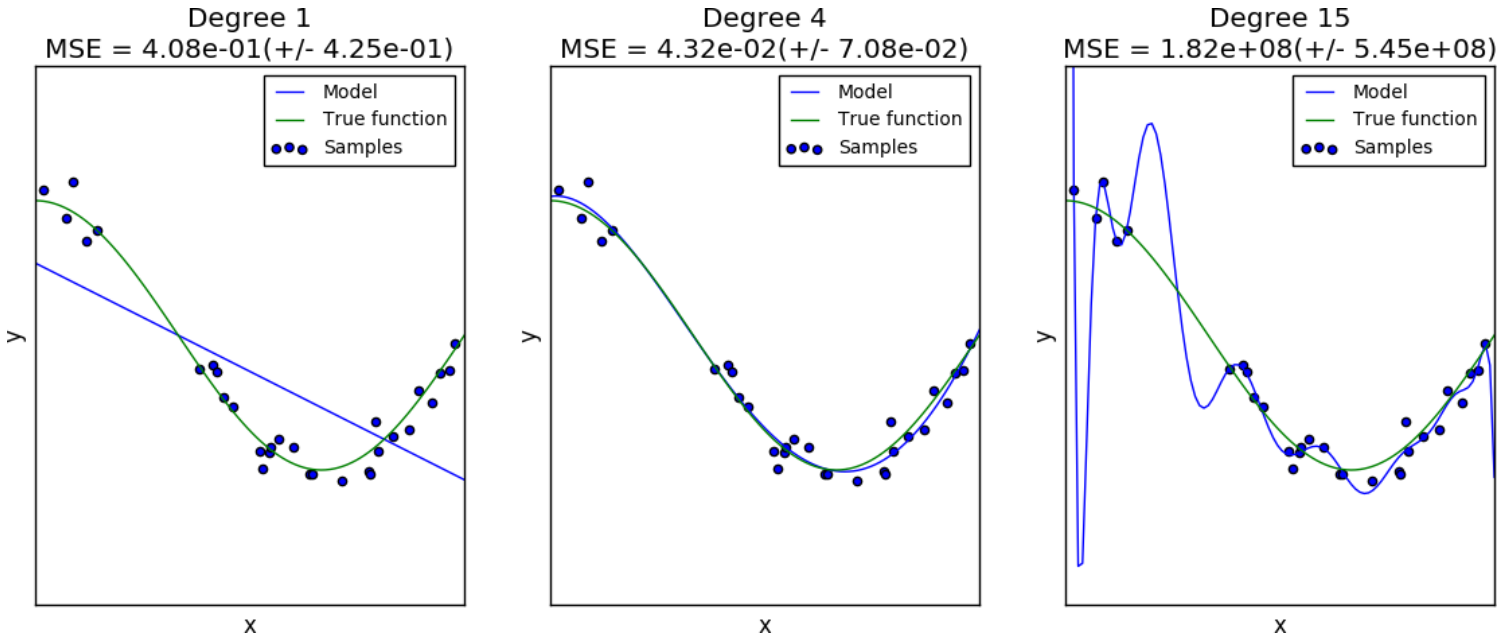
$$d = 1 \rightarrow n \geq 10 \textit{ data points}$$

$$d = 2 \rightarrow n \geq 10^2 = 100 \textit{ data points}$$

$$d = 3 \rightarrow n \geq 10^3 = 1000 \textit{ data points}$$

Problems

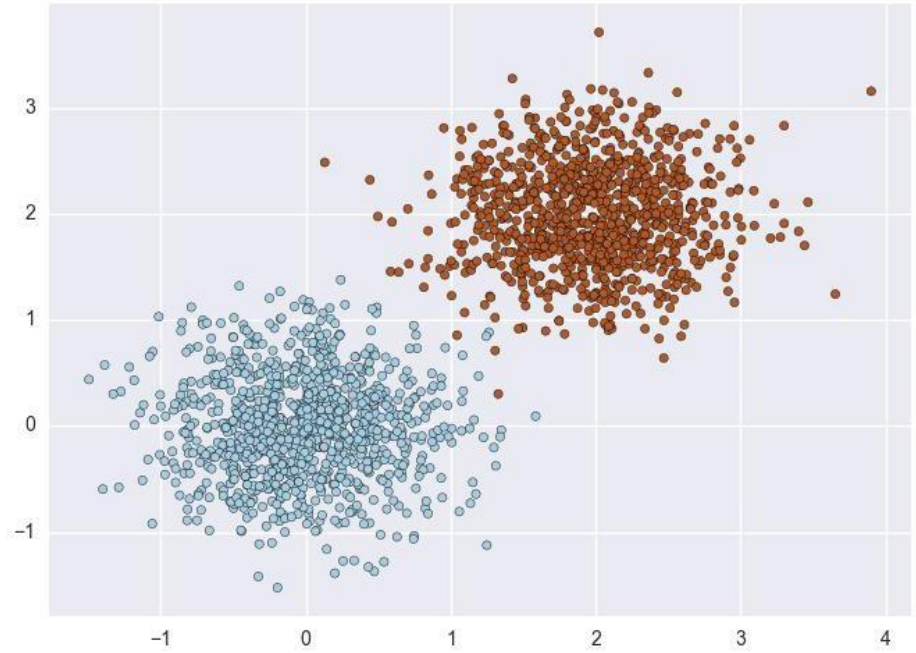
2) *Underfitting – Fair fitting* - Overfitting



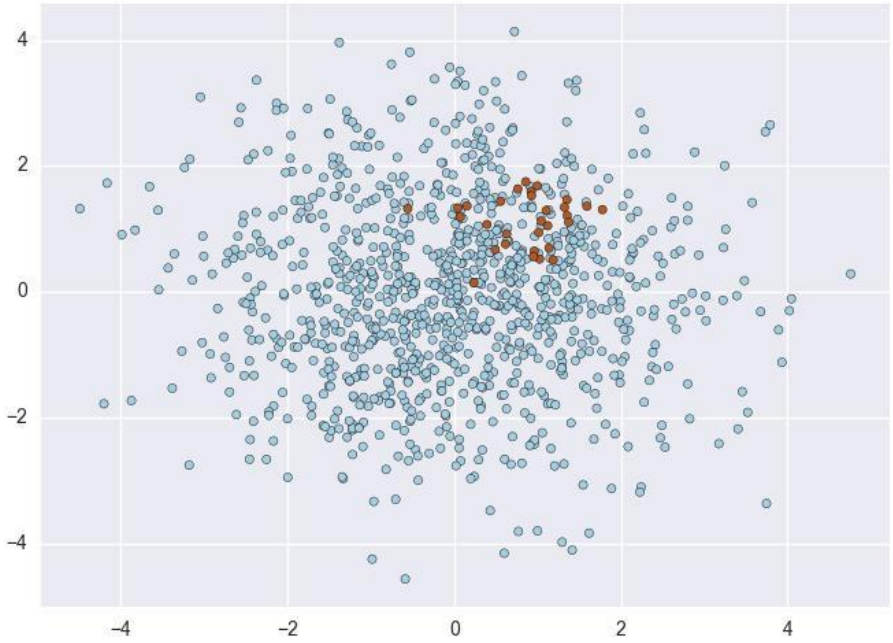
Problems

3) *Unbalance* and *overlapping*

BEST CASE SCENARIO



WORST CASE SCENARIO



Problems

Unbalance (I)

What can we do?

We can define new measures of "Accuracy" that are better behaved for unbalance.

$$G_m(\%) = \left[\sqrt{TP_r \cdot TN_r} \right] \cdot 100$$

$$A_{cw}(\%) = \{[\omega \cdot TP_r] + [(1-\omega) \cdot TN_r]\} \cdot 100$$

You can give more importance to "P" or to "N"

Here: $\boldsymbol{TP_r = \frac{TP}{TP+FN}}$; $\boldsymbol{TN_r = \frac{TN}{TN+FP}}$

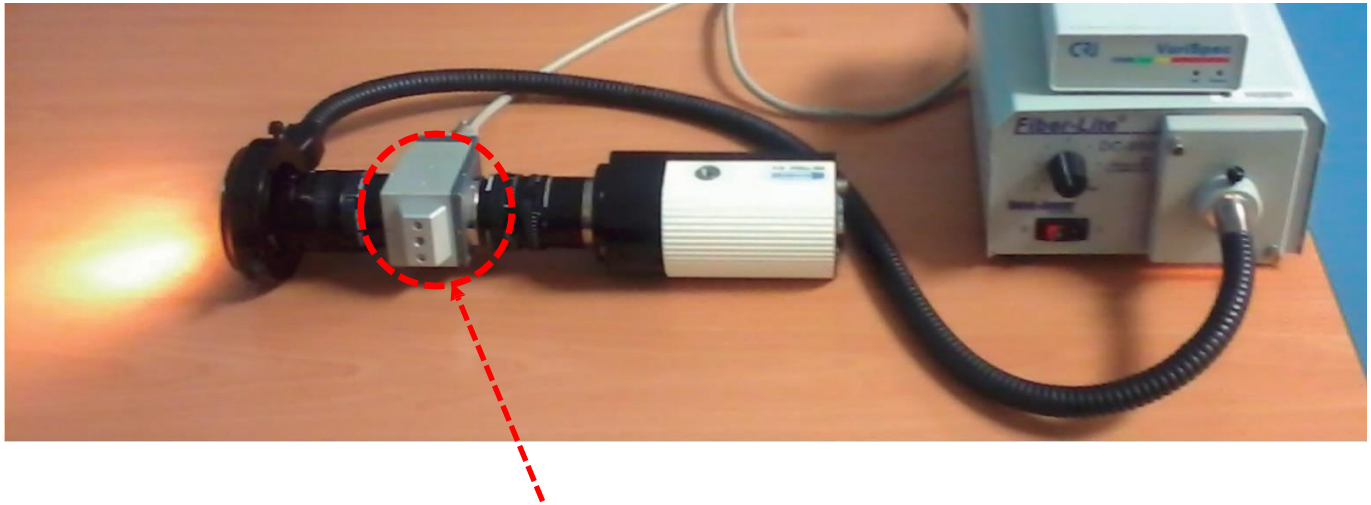
a. $\boldsymbol{0 \leq \omega \leq 1}$

b. $\boldsymbol{\omega + (1-\omega) = 1}$

Problems

Unbalance (II)

Imagine a ***melanoma*** detection problem. We acquire a multi-spectral image of a skin lesion.



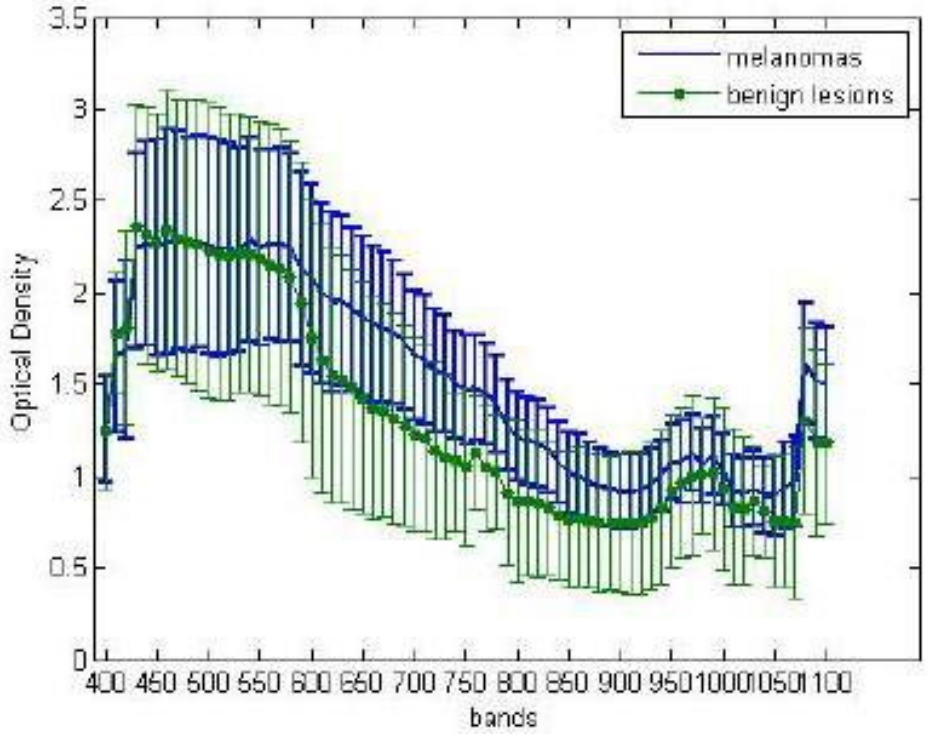
Liquid Crystal Tunable Filter (LCTF)

32 λs in $[400, 720]$nm; **32 λs** in $[650, 1100]$nm

Problems

Unbalance (III)

The spectral curves associated to melanoma and other skin lesions are different.



Problems

Unbalance (IV)

Imagine we have 520 patients, 20 of which suffer from melanoma.

Let's consider the following *confusion* matrices:

30 % of mistake
in the class *of interest!*

No, No, No !!!!!!!!

	"Melanoma"	No "Melanoma"
Classified as "Melanoma"	TP = 18	FP = 15
Classified as "No Melanoma"	FN = 2	TN = 485

	"Melanoma"	No "Melanoma"
Classified as "Melanoma"	TP = 14	FP = 15
Classified as "No Melanoma"	FN = 6	TN = 485

$$Acc\ (\%) = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) \cdot 100 =$$

$$= \left(\frac{18 + 485}{520}\right) \cdot 100 = \mathbf{96.73\ \%}$$

NO WAY!!

$$Acc\ (\%) = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) \cdot 100 =$$

$$= \left(\frac{14 + 485}{520}\right) \cdot 100 = \mathbf{95.96\ \%}$$

Problems

Unbalance (V)

Remember: 520 patients, 20 of which suffer from melanoma.

Let's consider the following *confusion* matrices:

	"Melanoma"	No "Melanoma"
Classified as "Melanoma"	TP = 18	FP = 15
Classified as "No Melanoma"	FN = 2	TN = 485

	"Melanoma"	No "Melanoma"
Classified as "Melanoma"	TP = 14	FP = 15
Classified as "No Melanoma"	FN = 6	TN = 485

$$TP_r = \frac{TP}{TP+FN}; \quad TN_r = \frac{TN}{TN+FP}$$

$$G_m(\%) = \left[\sqrt{TP_r \cdot TN_r}\right] \cdot 100 = \sqrt{\left(\frac{18}{20}\right) \cdot \left(\frac{485}{500}\right)} \cdot 100 = \mathbf{93.43}\ \%$$

BETTER BEHAVIOUR!!!

$$G_m(\%) = \left[\sqrt{TP_r \cdot TN_r}\right] \cdot 100 = \sqrt{\left(\frac{14}{20}\right) \cdot \left(\frac{485}{500}\right)} \cdot 100 = \mathbf{82.40}\ \%$$

Problems

Unbalance (VI)

Bear in mind: Not all mistakes might be **equally IMPORTANT!!!**

	"Melanoma"	No "Melanoma"
Classified as "Melanoma"	TP = 18	FP = 15
Classified as "No Melanoma"	FN = 2	TN = 485

	"Melanoma"	No "Melanoma"
Classified as "Melanoma"	TP = 14	FP = 15
Classified as "No Melanoma"	FN = 6	TN = 485

↑ NO. OF "FNs"

IS THE WORST MISTAKE OF THEM ALL IN THIS CASE

You are telling the patient he/she does not have melanoma but he/she has it!

References

1 Christopher Bishop, Pattern Recognition and Machine Learning, Information Science and Statistics Series, Springer Verlag, 2006.

2 Sergios Theodoridis, Konstantinos koutroumbas, Pattern Recognition, Academic press, 2009.