

# VC 13/14 – T17

## Pattern Recognition

Mestrado em Ciência de Computadores  
Mestrado Integrado em Engenharia de Redes e  
Sistemas Informáticos

***Miguel Tavares Coimbra***

# Outline

- Introduction to Pattern Recognition
- Statistical Pattern Recognition
- Classifiers

# Topic: Introduction to Pattern Recognition

- Introduction to Pattern Recognition
- Statistical Pattern Recognition
- Classifiers

This is a  
horse

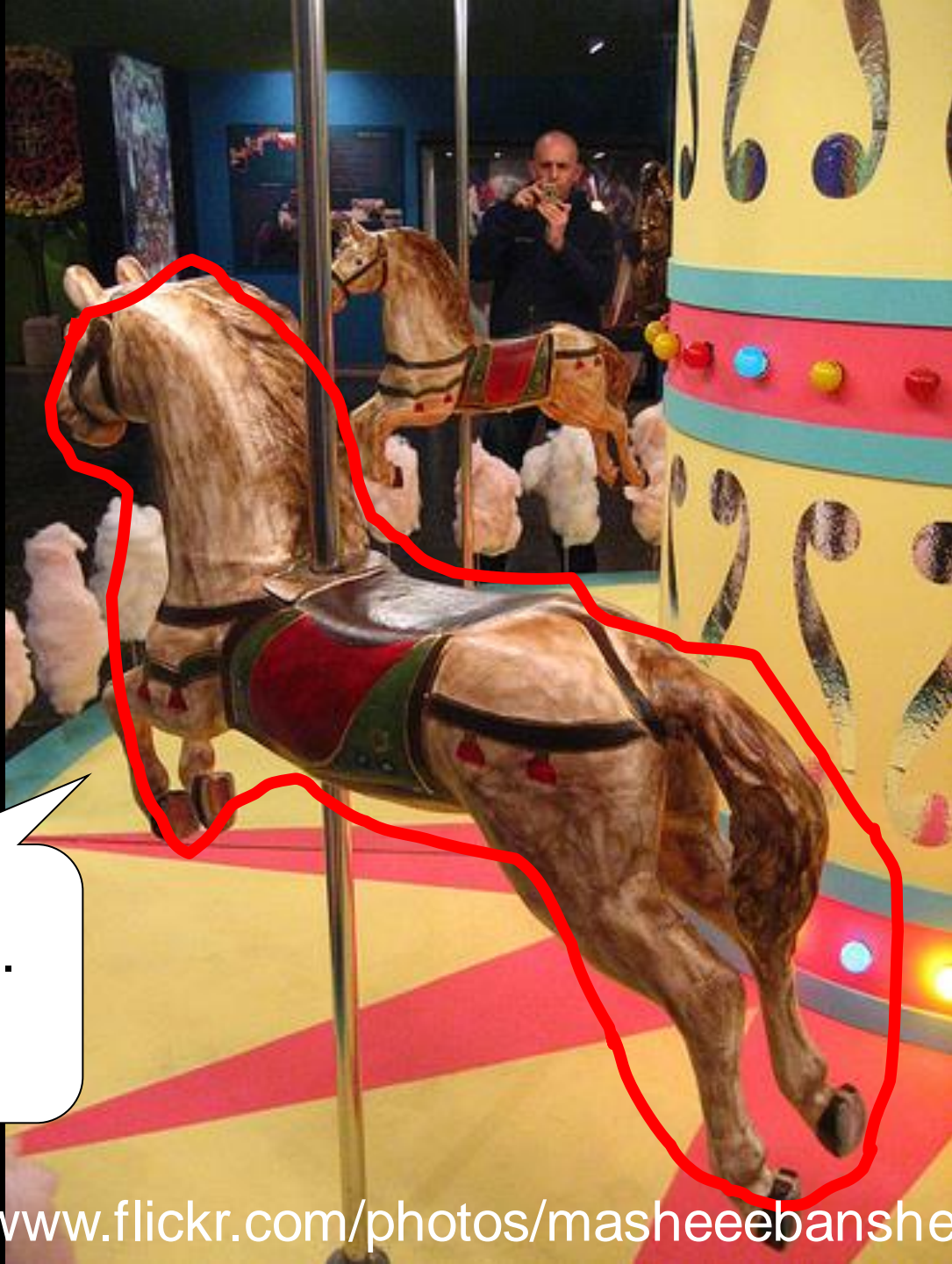


This is a  
horse



andrea.lindenberg © 2007

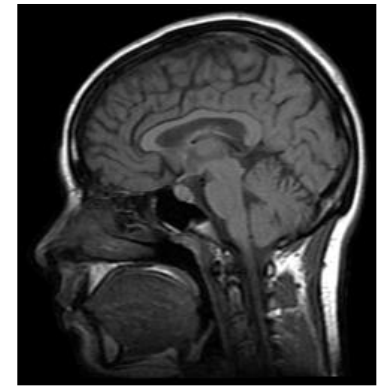
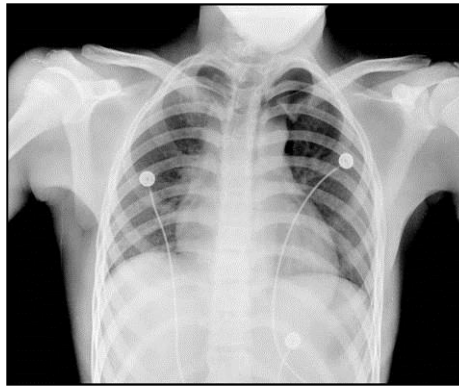
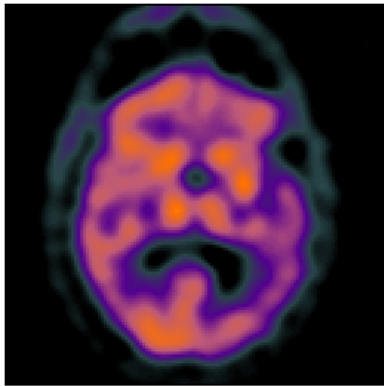
<http://www.flickr.com/photos/genewolf/2031802050/>



This is a...  
Horse?

# Decisions

- I can **manipulate** images.
- I want to make **decisions!**



- Classify / Identify **features**.
- Recognize **patterns**.

# One definition

- **Pattern recognition**

"the act of taking in raw data and taking an action based on the category of the data".

Wikipedia

- How do I do this so well?
- How can I make machines do this?



# The problem



Do you 'see'  
a horse?

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	2	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What a computer sees

# Mathematics

- **We only deal with numbers.**
  - How do we represent knowledge?
  - How do we represent visual features?
  - How do we classify them?
- **Very complex problem!!**
  - Let's break it into smaller ones...

# Typical PR system

## Sensor

Gathers the observations to be classified or described



## Feature Extraction

Computes numeric or symbolic information from the observations;



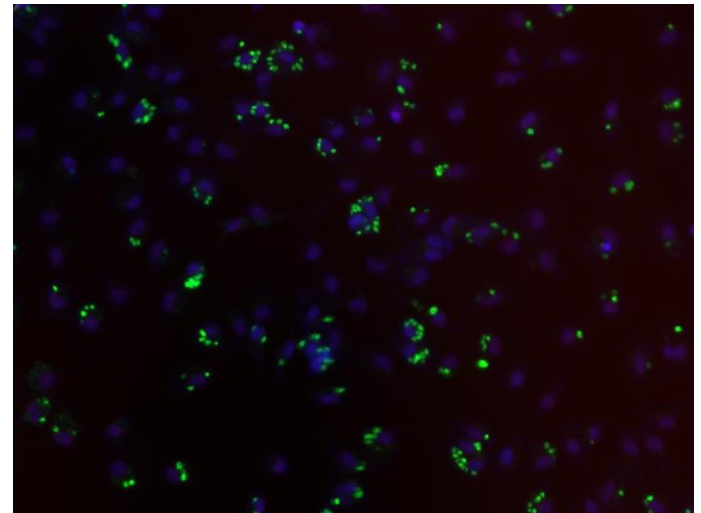
## Classifier

Does the actual job of classifying or describing observations, relying on the extracted features.

# Sensor

- In our specific case:
  - Image acquiring mechanism.
  - Let's assume we don't control it.

One observation = One Image  
Video = Multiple Observations



# Feature Extraction

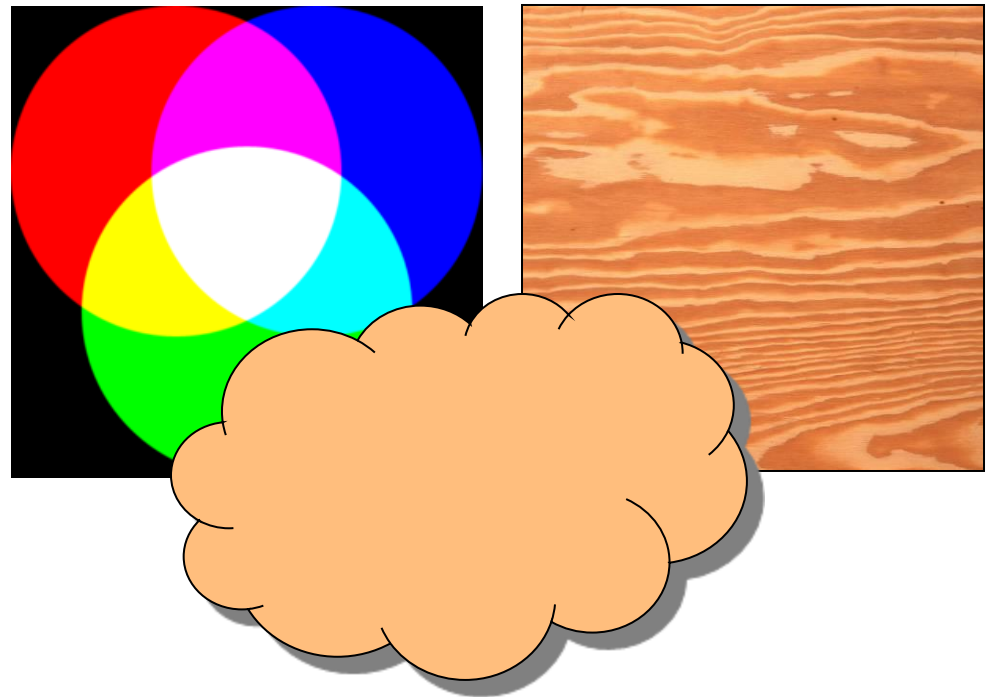
- **What exactly are features?**
  - Colour, texture, shape, etc.
  - Animal with 4 legs.
  - Horse.
  - Horse jumping.
- **These vary a lot!**
- **Some imply some sort of ‘recognition’**  
e.g. How do I know the horse is jumping?

# Broad classification of features

- **Low-level**
  - Colour, texture
- **Middle-level**
  - Object with head and four legs.
  - Object moving up.
  - Horse
- **High-level**
  - Horse jumping.
  - Horse competition.

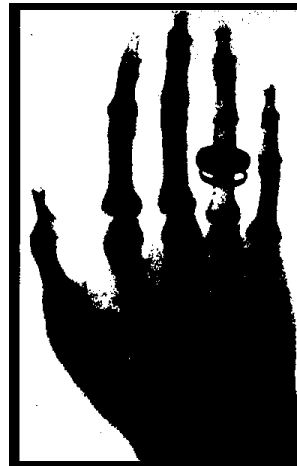
# Low-level features

- Objective
- Directly reflect specific image and video features.
  - Colour
  - Texture
  - Shape
  - Motion
  - Etc.



# Middle-level features

- Some degree of subjectivity
- They are typically one solution of a problem with multiple solutions.
- Examples:
  - Segmentation
  - Optical Flow
  - Identification
  - Etc.





# High-level features

- Semantic Interpretation
- Knowledge
- Context
- Examples:



How do humans do this so well?

- This person suffers from epilepsy.
- The virus attacks the cell with some degree of intelligence.
- This person is running from that one.

# The semantic gap

- Fundamental problem of current research!

## Low-level:

- Colour
- Texture
- Shape
- ...



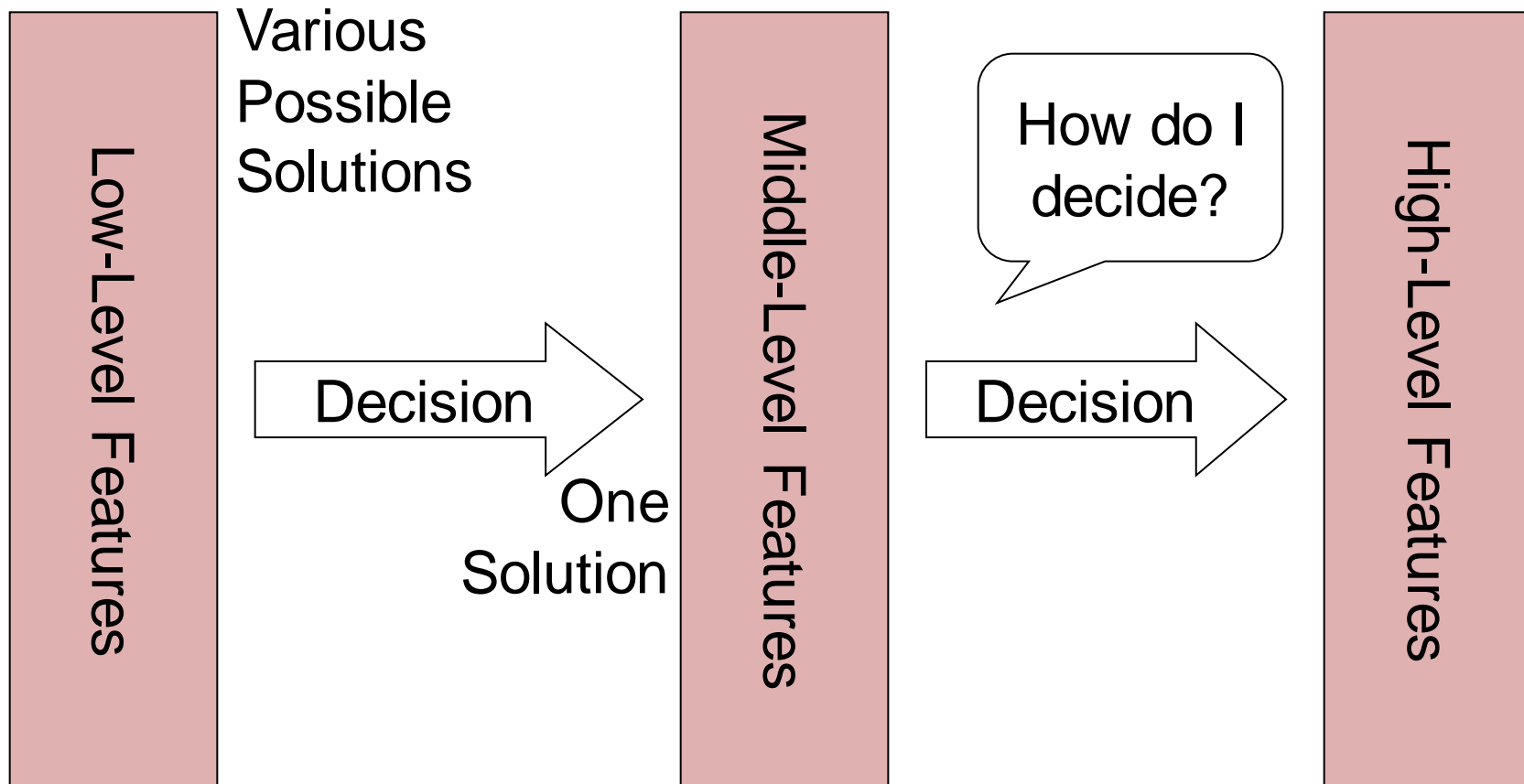
## High-level:

- Interpretation
- Decision
- Understanding
- ...

Now what??  
How do i cross this  
bridge?



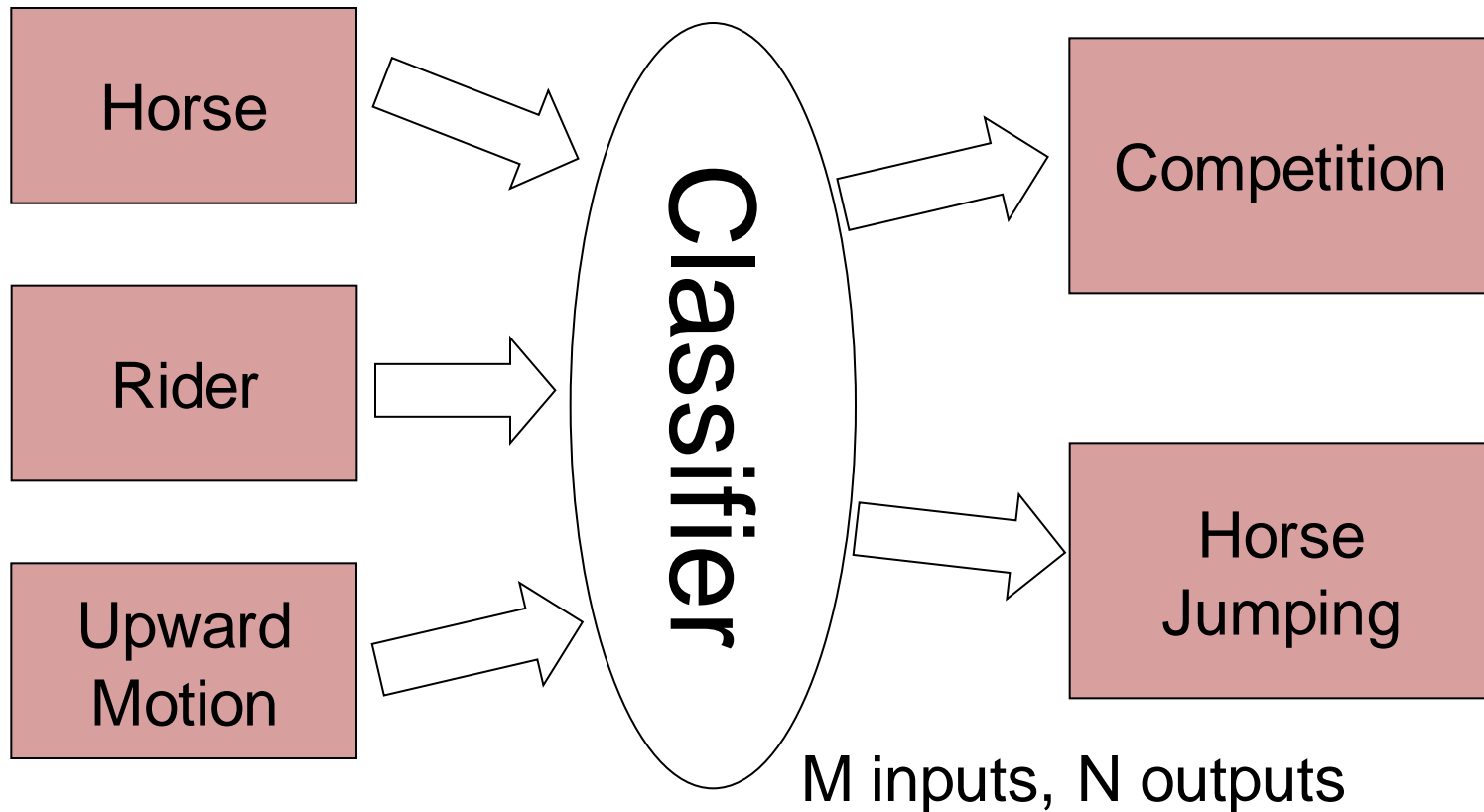
# Features & Decisions



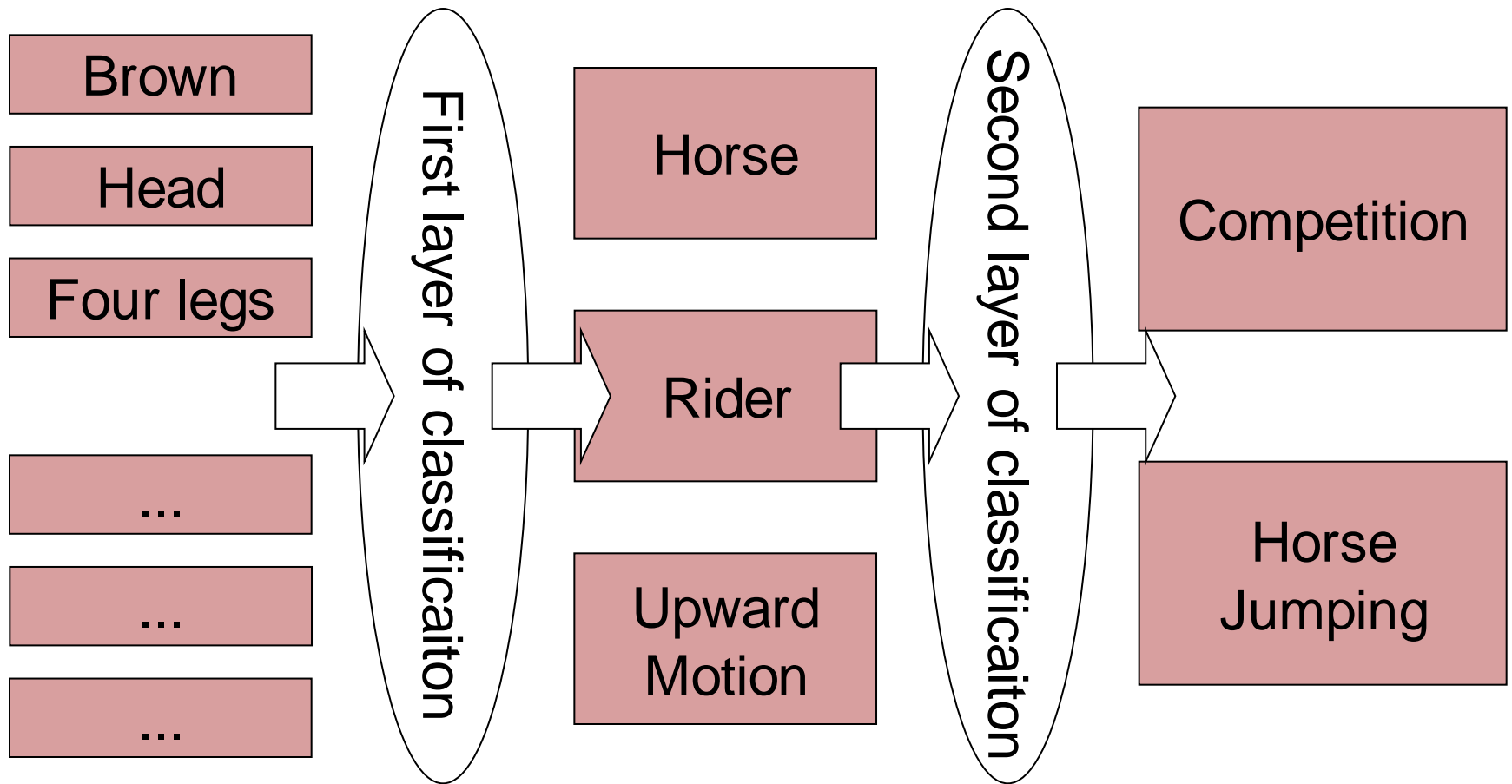
# Classification

Middle-Level Features

High-Level Features



# Layers of classification



# Classifiers

- How do I map my  $M$  inputs to my  $N$  outputs?
- Mathematical tools:
  - Distance-based classifiers.
  - Rule-based classifiers.
  - Neural Networks.
  - Support Vector Machines
  - ...

# Types of PR methods

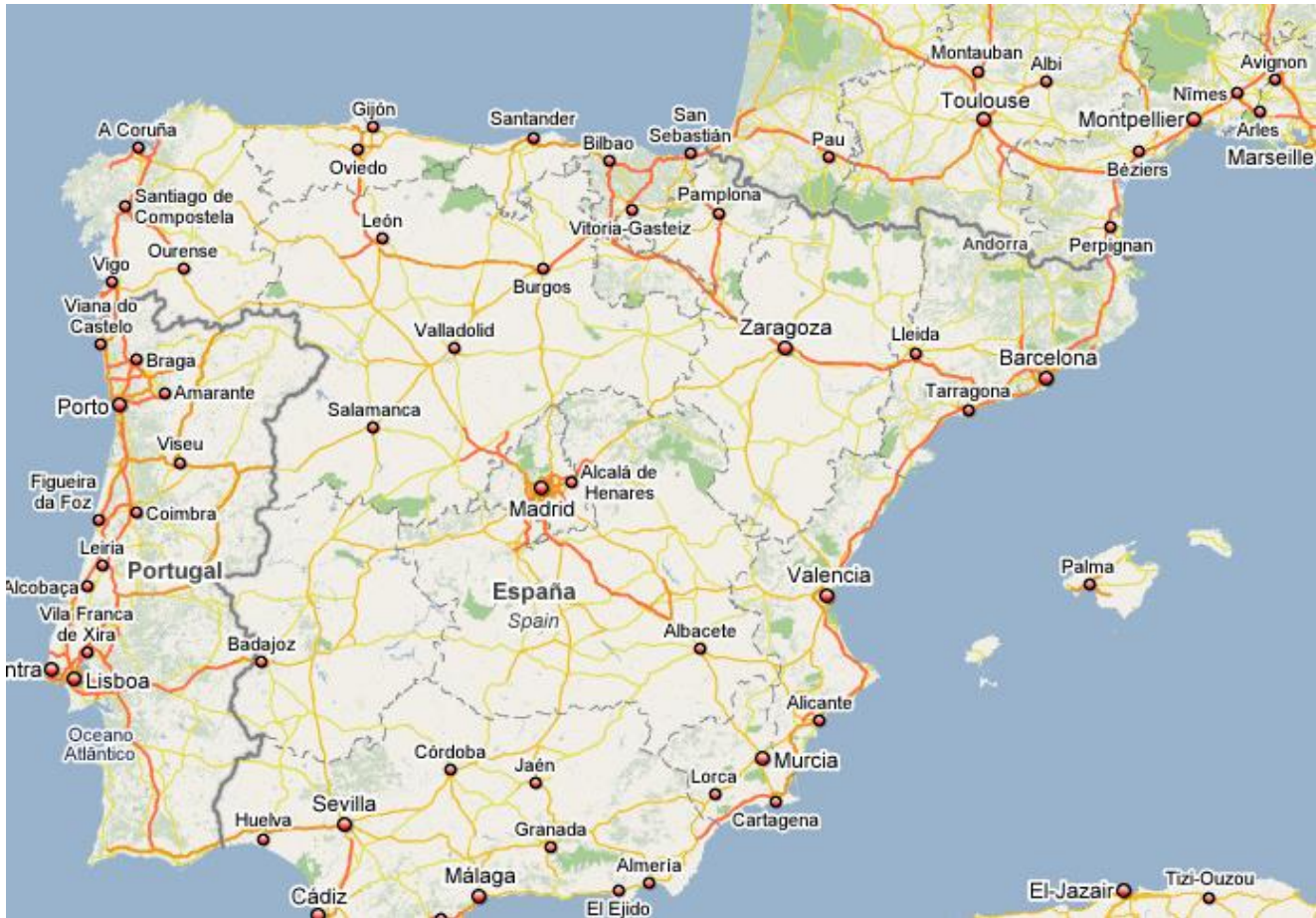
- **Statistical pattern recognition**
  - based on statistical characterizations of patterns, assuming that the patterns are generated by a probabilistic system.
- **Syntactical (or structural) pattern recognition**
  - based on the structural interrelationships of features.

# Topic: Statistical Pattern Recognition

- Introduction to Pattern Recognition
- **Statistical Pattern Recognition**
- Classifiers



# Is Porto in Portugal?



# Porto is in Portugal

- I want to make decisions.
  - Is Porto in Portugal?
- I know certain things.
  - A world map including cities and countries.
- I can make this decision!
  - Porto is in Portugal.
- I had enough *a priori* knowledge to make this decision.

# What if I don't have a map?

- I still want to make this decision.
- I observe:
  - Amarante has coordinates  $x_1, y_1$  and is in Portugal.
  - Viseu has coordinates  $x_2, y_2$  and is in Portugal.
  - Vigo has coordinates  $x_3, y_3$  and is in Spain.
- I classify:
  - Porto is close to Amarante and Viseu so **Porto is in Portugal.**
- What if I try to classify *Valença*?

# Statistical PR

- I used **statistics** to make a decision.
  - I can make **decisions** even when I don't have full a priori knowledge of the whole process.
  - I can make **mistakes**.
- How did I **recognize** this pattern?
  - I **learned** from previous observations where I knew the classification result.
  - I **classified** a new observation.

What pattern?

# Back to the Features

- Feature  $F_i$       $F_i = [f_i]$

- Feature  $F_i$  with  $N$  values.

$$F_i = [f_{i1}, f_{i2}, \dots, f_{iN}]$$

- Feature vector  $F$  with  $M$  features.

$$F = [F_1 | F_2 | \dots | F_M]$$

- Naming conventions:
  - Elements of a **feature vector** are called **coefficients**.
  - **Features** may have one or more **coefficients**.
  - **Feature vectors** may have one or more **features**.

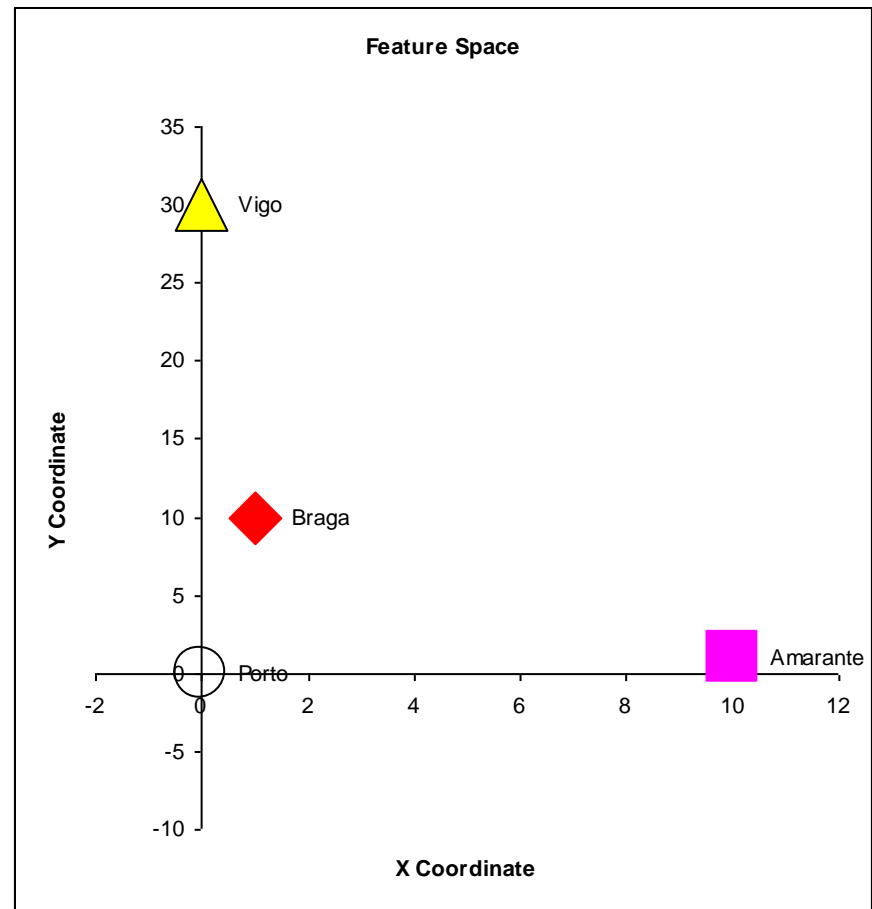
# Back to our Porto example

- I've classified that Porto is in Portugal.
- What feature did I use?
  - Spatial location
- Let's get more formal
  - I've defined a feature vector  $F$  with one feature  $F_1$ , which has two coefficients  $f_{1x}$ ,  $f_{1y}$ .

$$F = [F_1] = [f_{1x}, f_{1y}]$$

# Feature Space

- **Feature Vector**
  - Two total coefficients.
  - Can be seen as a feature 'space' with two orthogonal axis.
- **Feature Space**
  - Hyper-space with  $N$  dimensions where  $N$  is the total number of coefficients of my feature vector.



# A *Priori* Knowledge

- I have a precise **model** of my feature space based on **a priori** knowledge.

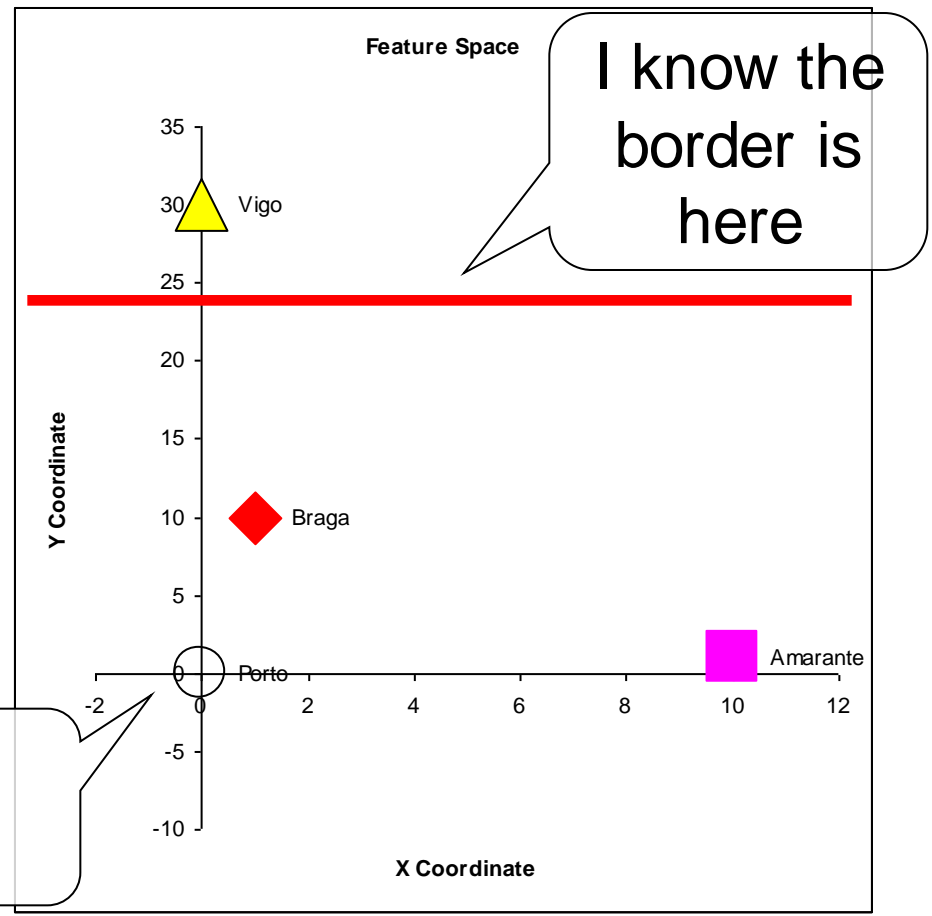
*City is in Spain if  $F_{1Y} > 23$*

- Great models = Great classifications.

*$F_{1Y}(\text{London}) = 100$*

*London is in Spain (??)*

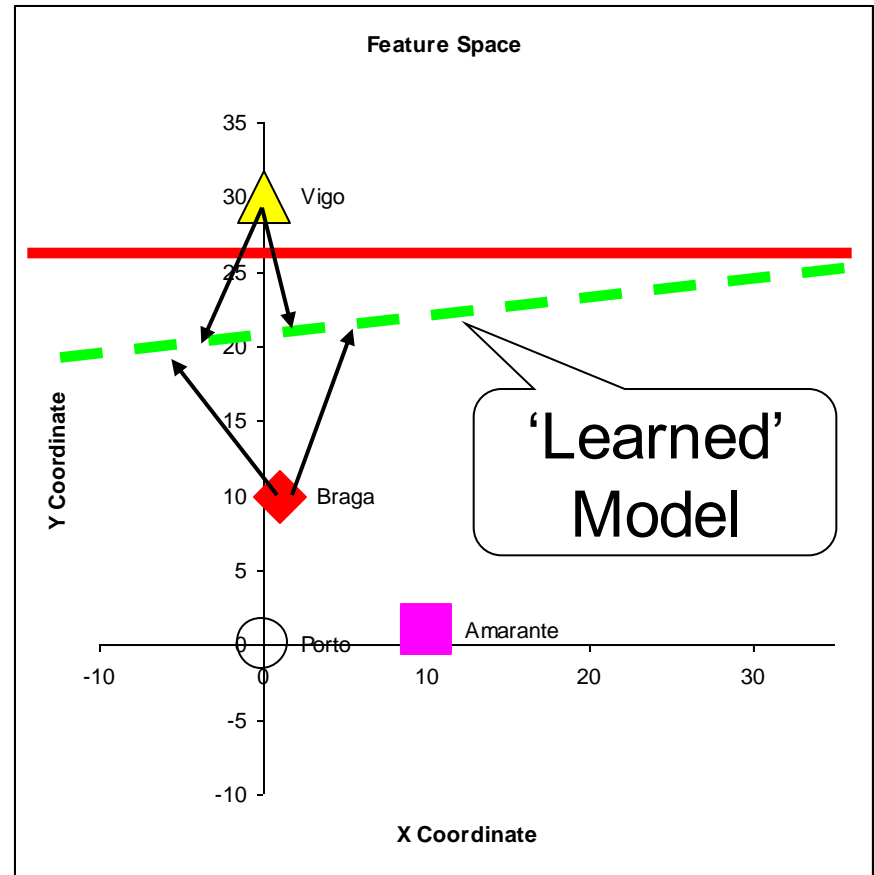
Porto **is** in Portugal!





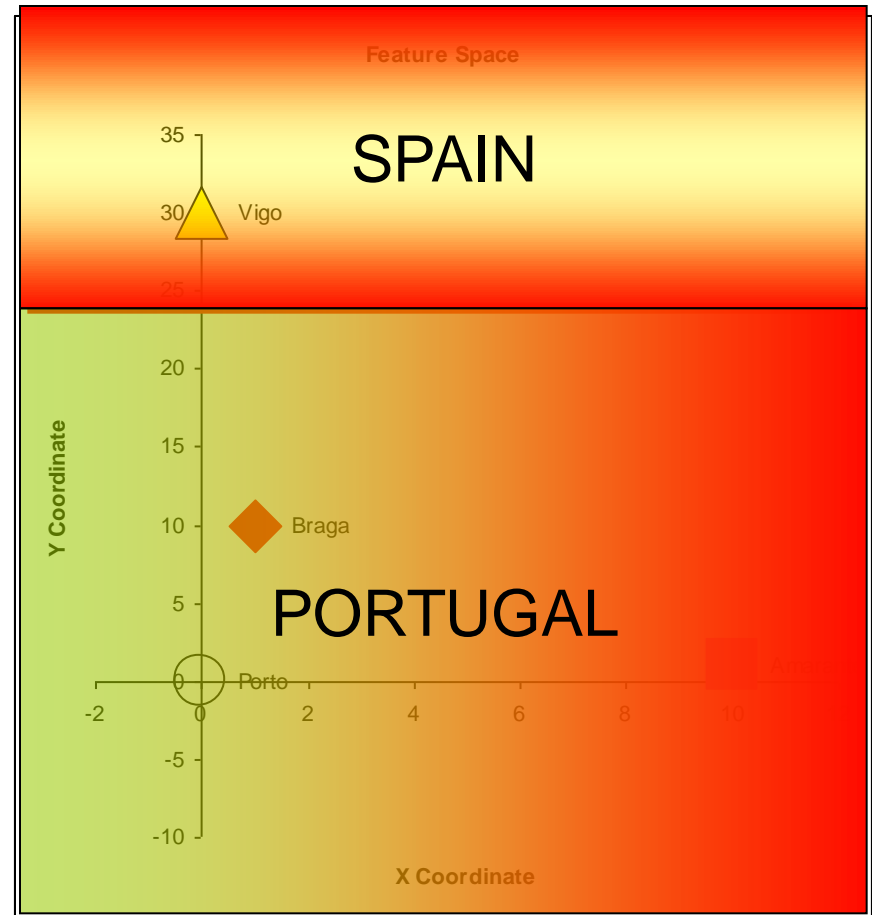
# What if I don't have a model?

- I need to **learn** from observations.
  - Derive a model.
  - Direct classification.
- **Training stage.**
  - Learn model parameters.
- **Classification**



# Classes

- In our example, cities can belong to:
  - Portugal
  - Spain
- I have two **classes** of cities.
- A **class** represents a sub-space of my feature space.



# Classifiers

- A **Classifier C** maps a class into the feature space.

$$C_{\text{Spain}}(x, y) = \begin{cases} \textit{true} & , y > K \\ \textit{false} & , \textit{otherwise} \end{cases}$$

- Various types of classifiers.
  - Nearest-Neighbours.
  - Bayesian.
  - Soft-computing machines.
  - Etc...

# Topic: Classifiers

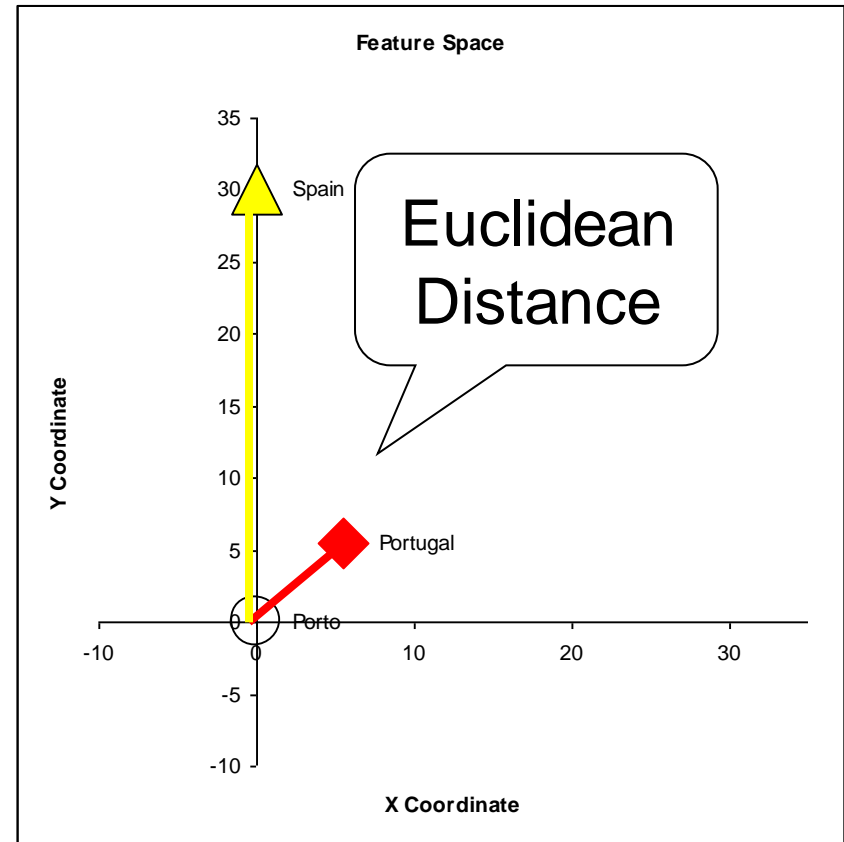
- Introduction to Pattern Recognition
- Statistical Pattern Recognition
- **Classifiers**

# Distance to Mean

- I can represent a class by its mean feature vector.

$$C = \bar{F}$$

- To classify a new object, I choose the class with the closest mean feature vector.
- Different distance measures!



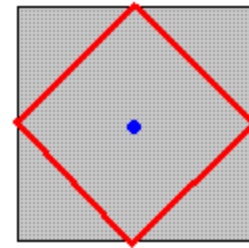
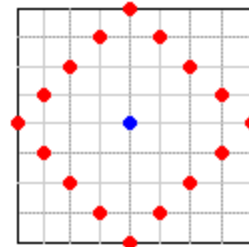
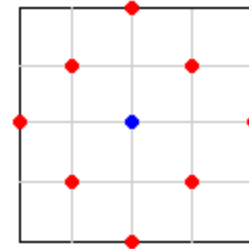
# Possible Distance Measures

- L1 Distance

$$L1 = \frac{1}{N} \sum_{x=1}^N |S(x) - v(x)|$$

- Euclidean Distance  
(L2 Distance)

$$L2 = \frac{1}{N} \sum_{x=1}^N (S(x) - v(x))^2$$

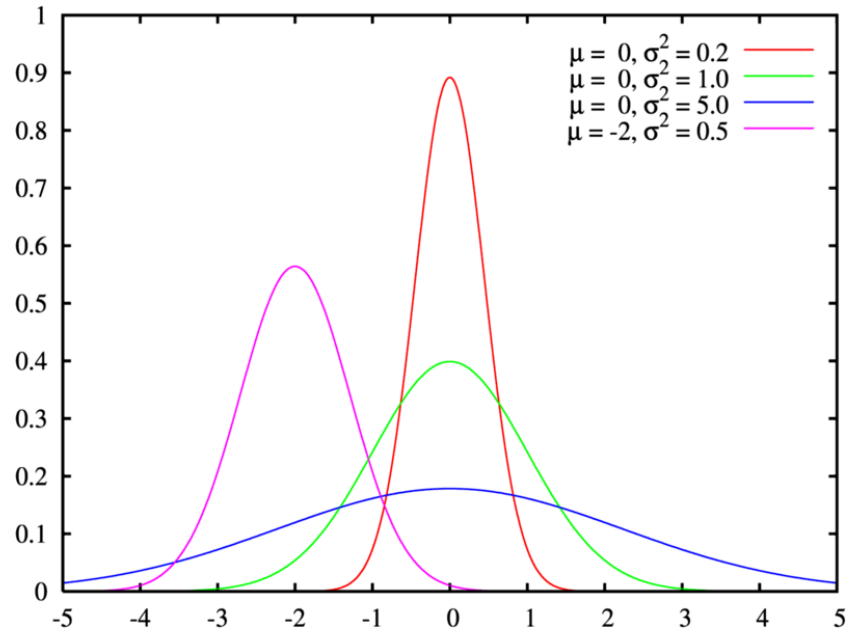


L1 or  
Taxicab  
Distance

# Gaussian Distribution

- Defined by two parameters:
  - Mean:  $\mu$
  - Variance:  $\sigma^2$
- Great approximation to the distribution of many phenomena.
  - *Central Limit Theorem*

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$



# Multivariate Distribution

- For N dimensions:

$$f_X(x_1, \dots, x_N) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu)\right)$$

- Mean feature vector:

$$\mu = \bar{F}$$

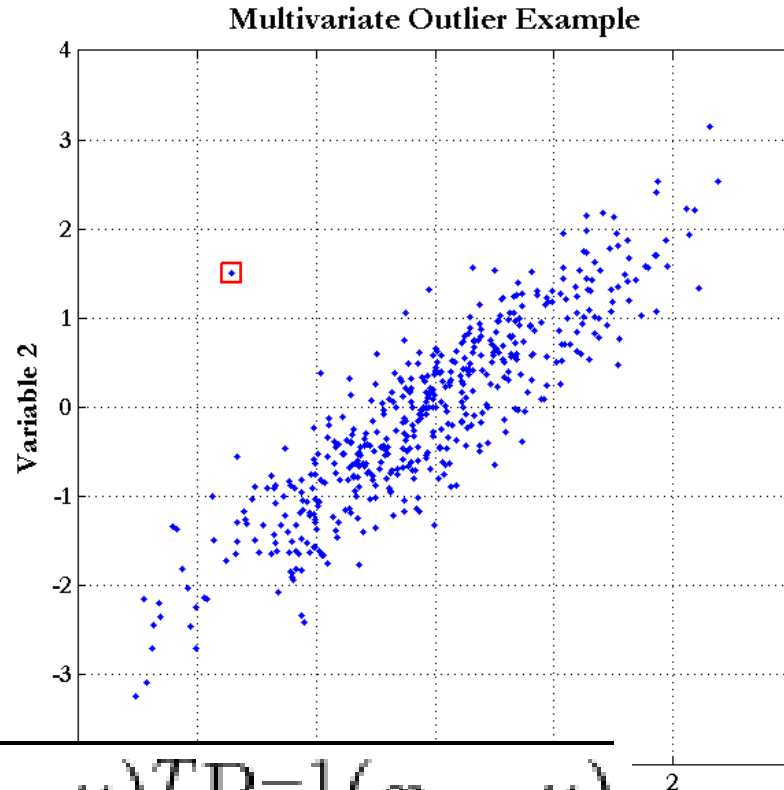
- Covariance Matrix:

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \quad \mu_i = \mathbb{E}(X_i) \quad \Sigma_{ij} = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]$$



# Mahalanobis Distance

- Based on the covariance of coefficients.
- Superior to the Euclidean distance.



$$D_M(x) = \sqrt{(x - \mu)^T P^{-1} (x - \mu)}.$$

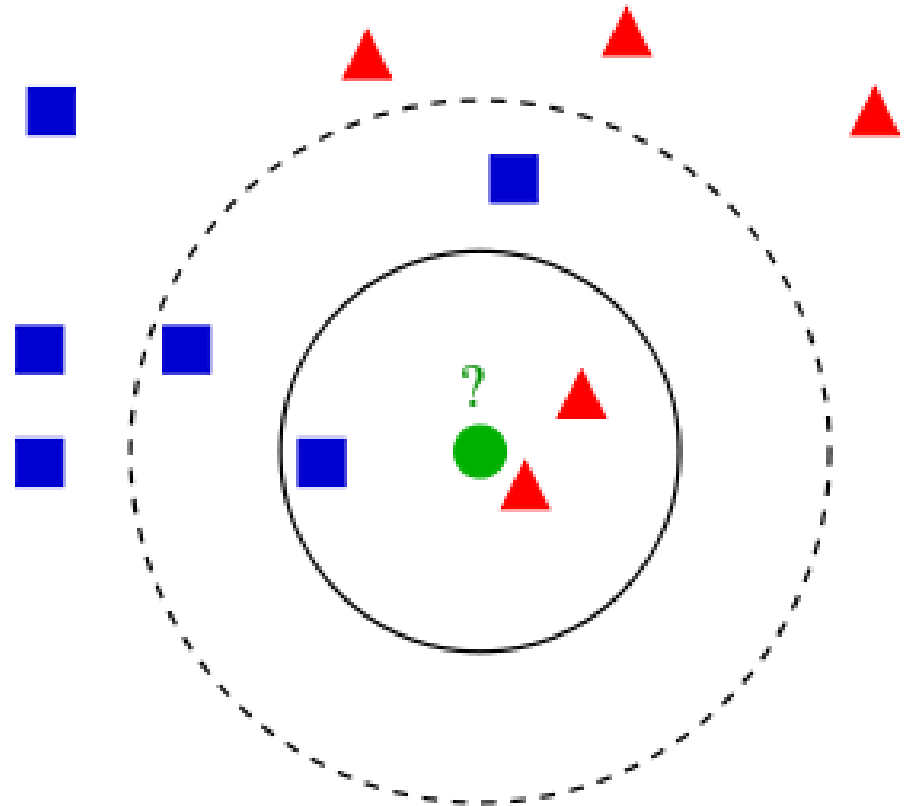
# K-Nearest Neighbours

- **Algorithm**

- Choose the closest K neighbours to a new observation.
- Classify the new object based on the **class** of these K objects.

- **Characteristics**

- Assumes no model.
- Does not scale very well...



# Resources

- Gonzalez & Woods, 3rd Ed, Chapter 12.
- Andrew Moore, Statistic Data Mining Tutorial, <http://www.autonlab.org/tutorials/>