ENCARGO DE PROGRAMA

ENCARGO

TAREA DE PROGRAMACIÓN

- **Quiero un programa** que dado los datos académicos de un alumno me calcule si la nota final es aprobado o no.
- Quiero hacerlo como se hizo en otros años, pero no me dijeron cuál era el criterio
- Lo que tengo son los datos de esos años pasados

ENCARGO

TAREA DE PROGRAMACIÓN

- Lo que tengo son los datos de esos años pasados

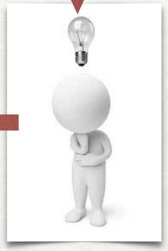
nombre	teoria	practicas	final
a1	false	false	false
a2	false	true	false
a3	true	true	true
...	...	...	..

ENCARGO

TAREA DE PROGRAMACIÓN

- Lo que pensó un programador llamado **OneR**

nombre	teoria	practicas	final
a1	false	false	false
a2	false	true	false
a3	true	true	true
...	...	...	..



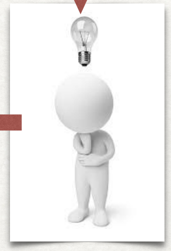
Program xP41_OneR
…
If nombre = a1, return false
if nombre = a2, return false
if nombre = a3, return true
…

ENCARGO

TAREA DE PROGRAMACIÓN

- Lo que pensó **C4.5**

nombre	teoria	practicas	final
a1	false	false	false
a2	false	true	false
a3	true	true	true
...	...	...	..



Program xP41_C4.5
…
If teoria & practica, return true
else return false

…

ENCARGO

TAREA DE PROGRAMACIÓN

- Lo que pensó **Lineal**

nombre	teoria	practicas	final
a1	false	false	false
a2	false	true	false
a3	true	true	true
...	...	...	..

nombre	notas
a1	(0,0)
a2	(0,1)
a3	(1,1)
...	..

Nota	n1	n2
peso	w1	w2

Producto escalar = n1*w1 + n2*w2

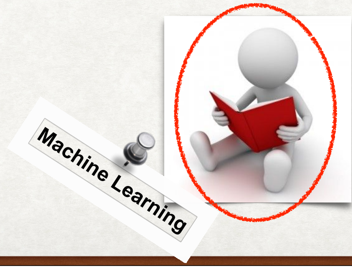
Program xP41_lineal
…
w1, w2 = 1
return
sign(producto escalar-1.5)
…

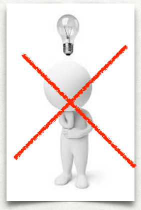
CRISIS

PRESCINDIR DEL PROGRAMADOR

- Medir la calidad de los programas que hacen

- Contratar a un nuevo programador que haga un programa que devuelva el programa que hubiese hecho el mejor programador





ENCARGO

TAREA DE PROGRAMACIÓN

Evaluación

nombre	teoria	practicas	final
a1	false	false	false
a2	false	true	false
a3	true	true	true
...	...	...	..

nombre	teoria	practicas	final
A100	false	True	false
A101	true	true	true
A102	true	true	true
...	...	...	..

nombre	teoria	practicas
A100	false	True
A101	true	true
A102	true	true
...	...	...

Program xP41_C4.5
...
If teoria & practica, return true
else return false

...

final Pre
false
true
true
..

=?

final Real
false
true
true
..

SYLLABUS

- Input data

nombre	teoria	practicas
a1	false	false
a2	false	true
a3	true	true
...	...	...

final
false
false
true
..

- Output data (class ...)
- Automatic programming
- More data than knowledge

Acknowledgements

Most slides come from

Machine Learning Techniques for Data Mining

Eibe Frank
University of Waikato
New Zealand

Introduction to Machine Learning

Oscar Luaces

Index

Induction as a tool for knowledge acquisition

1. Bibliography and laboratory software
2. The conceptual problem of Machine Learning or pattern analysis
3. Pattern recognition tasks

Bibliography and laboratory software

Ian H. Witten, Eibe Frank

Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)

Morgan Kaufmann, 2005

- The material provided in this course

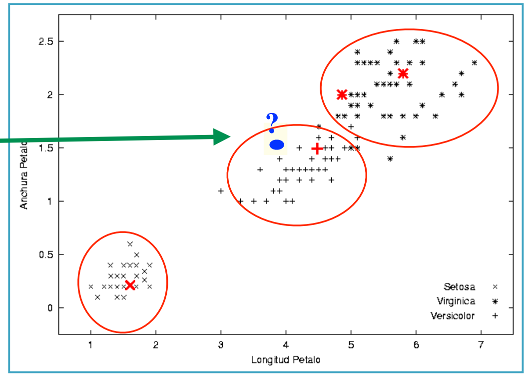
WEKA The University of Waikato

http://www.cs.waikato.ac.nz/ml/weka/

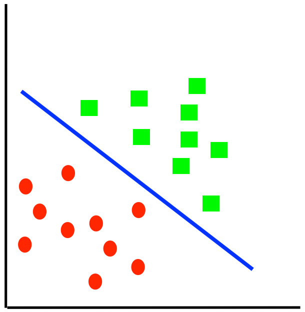
The conceptual problem of Machine Learning: training examples

Classification

Which class of iris am I?

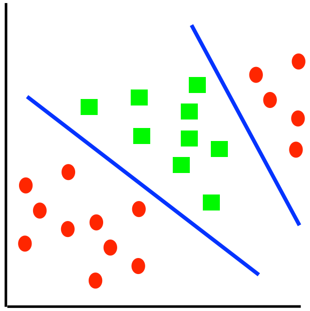


Linearly separable



Weight	Length	Class
61.2	172.8	Yes
54.3	165.5	No
77.4	182.3	Yes
81.2	175.3	Yes
57.7	162.2	No

Linearly not separable



Weight	Length	Class
61.2	172.8	Yes
54.3	165.5	No
87.2	182.3	No
81.2	175.3	Yes
57.7	162.2	No

Pattern recognition tasks

Searching for regularities on data

- Supervised (classes): **Induction of functions**:

  classification (class discrete values/nominal)
  regression (class continuous (ordinal) values)
  ranking (learning to order)

- Unsupervised (no classes): **Clustering**

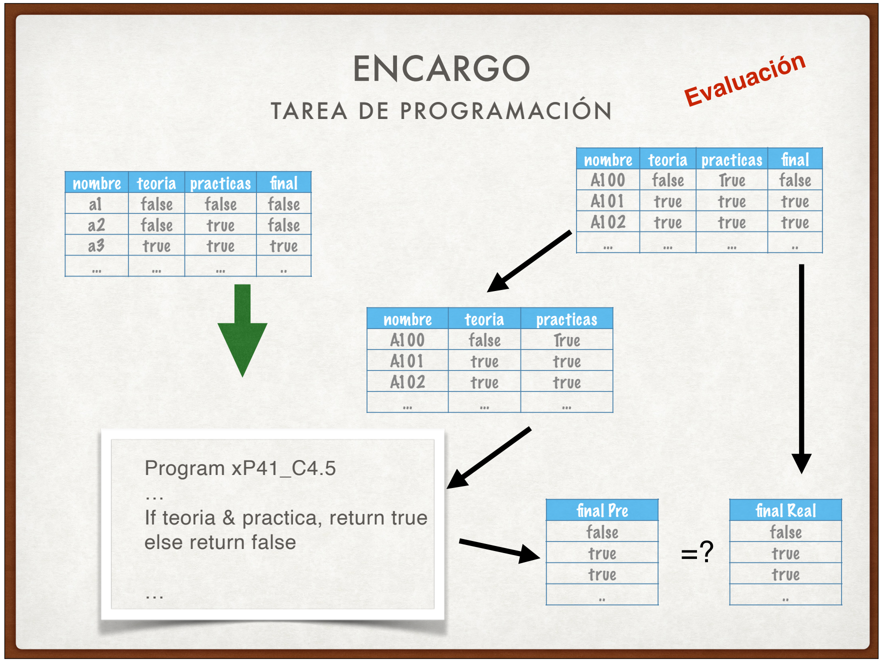
- Dimensionality reduction
  (for graphical representations, but not only)

INDUCTIVE LEARNING
FORMAL FRAMEWORK

- From a dataset

A_1	A_2	...	A_n	Pred.
$V_{1,1}$	$V_{1,2}$	...	$V_{1,n}$	C_1
$V_{2,1}$	$V_{2,2}$	...	$V_{2,n}$	C_2
...	...	...	...	...
$V_{m,1}$	$V_{m,2}$	...	$V_{m,n}$	C_m

- We aim is to obtain a **hypothesis** $h \in H$, such that assigns the *right* class from the other attribute values (A_1, A_2, ...)



Right predictions = correct in unseen cases

Given

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$

where

$x_i \in X$ input space; $y_i \in Y$ output space (classes)

The aim, a hypothesis

$$h : X \longrightarrow Y$$

(in a space of hypotheses: linear, polynomial, ...)

such that

minimize the expected **loss** over all test sets S' drawn from the same distribution of S (independent and identically distributed i.i.d.)

Loss functions

$$S' = \{(x'_1, y'_1), \ldots, (x'_n, y'_{n'})\}$$

Classification

$$\Delta((h(x'_1), \ldots, h(x'_{n'})), (y'_1, \ldots, y'_{n'})) = \frac{1}{n'} \sum_{i=1}^{n'} 1_{(h(x'_i) \neq y'_i)}$$

Regression

$$\Delta((h(x'_1), \ldots, h(x'_{n'})), (y'_1, \ldots, y'_{n'})) = \frac{1}{n'} \sum_{i=1}^{n'} |h(x'_i) - y'_i|$$

$$\Delta((h(x'_1), \ldots, h(x'_{n'})), (y'_1, \ldots, y'_{n'})) = \frac{1}{n'} \sum_{i=1}^{n'} (h(x'_i) - y'_i)^2$$

Classifiers: Simplicity first

- Simple algorithms often work very well!
- There are many kinds of simple structure, eg:
  - One attribute does all the work
  - Instance-based: use a few prototypes
- Success of method depends on the domain

Classifiers: Simplicity first

Zero R

- Computes the most frequent class in training set
- In testing always predict that class
- It uses **zero** attribute values

Classifiers: Simplicity first

Zero R

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

In **ANY** case
Play? Yes

Simplicity: 1 R (OneR)

- 1R: learns a 1-level decision tree
  - I.e., rules that all test one particular attribute
- Basic version
  - One branch for each value
  - Each branch assigns most frequent class
  - Error rate: proportion of instances that don't belong to the majority class of their corresponding branch
  - Choose attribute with lowest error rate

(Assumes nominal attributes)

Evaluating the weather attributes

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Attribute	Rules	Errors	Total errors
Outlook	Sunny → No	2/5	**4/14**
	Overcast → Yes	0/4	
	Rainy → Yes	2/5	
Temp	Hot → No*	2/4	5/14
	Mild → Yes	2/6	
	Cool → Yes	1/4	
Humidity	High → No	3/7	**4/14**
	Normal → Yes	1/7	
Windy	False → Yes	2/8	5/14
	True → No*	3/6	

- * indicates a tie

Instance based learning

- Simplest form of learning: rote learning
  - Training instances are searched for instances that most closely resembles new instance
  - The instances themselves represent the knowledge
  - Also called instance based learning

Similarity function defines what's "learned"

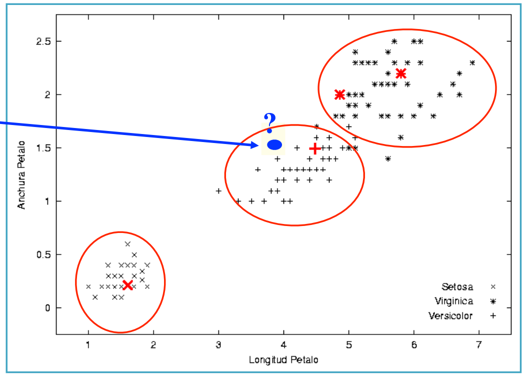
Instance based learning is *lazy* learning

Methods: nearest neighbor (NN),
k-nearest neighbor (kNN)

An example of classification by neighborhood

Classification

Which class of iris am I?



Instance based learning:
The distance function

- Simplest case: one numeric attribute
  - Distance is the difference between the two attribute values involved

  Several numeric attributes: normally, Euclidean distance is used and attributes are normalized

  Nominal attributes: distance is set to 1 if values are different, 0 if they are equal

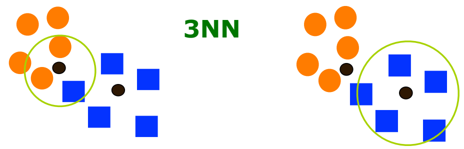
  Are all attributes equally important?
  - Weighting the attributes might be necessary

Nearest Neighbor variants: k-NN

The most prudent neighbor asks to more (**k**) people

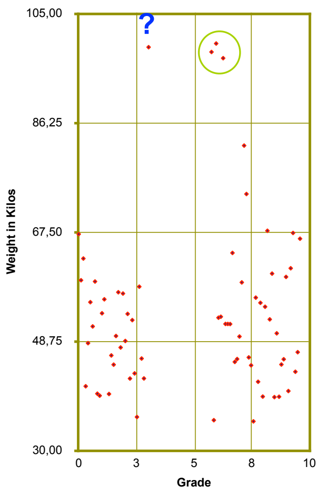
Saves all training set

To classify new data
- searches for the **k** nearest training sets to the new data
- predicts the most **frequent** class in these k examples



Neighborhood, problems with metrics

- Mess numeric and nominal values
- Relevance of attributes



Lo que se hace en 2022

Función de pérdida

$$\hat{y} = h_\theta(\mathbf{x})$$

$$\theta \leftarrow \underset{\theta}{\operatorname{argmin}} \frac{1}{|D|} \sum_{(x,y)\in D} \operatorname{loss}(y, h_\theta(\boldsymbol{x}))$$