Artificial Intelligence
EDA132
Lecture 7: Machine Learning

Pierre Nugues

Lund University
Pierre.Nugues@cs.lth.se
http://cs.lth.se/pierre_nugues/

February 11, 2014

Why Machine Learning: Early Artificial Intelligence

Early artificial intelligence techniques used introspection to codify knowledge, often in the form of rules.
Expert systems, one of the most notable applications of traditional AI, were entirely based on the competence of experts.
This has two major drawbacks:

- Need of an expertise to understand and explain the rules
- Bias introduced by the expert

Why Machine Learning: What has Changed

Now terabytes of data available.

Makes it impossible to understand such volumes of data, organize them using manually-crafted rules.

Triggered a major move to empirical and statistical techniques.

In fact, most machine–learning techniques come from traditional statistics.

Applications in natural language processing, medicine, banking, online shopping, image recognition, etc.

> *The success of companies like Google, Facebook, Amazon, and Netflix, not to mention Wall Street firms and industries from manufacturing and retail to healthcare, is increasingly driven by better tools for extracting meaning from very large quantities of data. 'Data Scientist' is now the hottest job title in Silicon Valley.*

– Tim O'Reilly

Some Definitions

Machine learning always starts with **data sets**: a collection of objects or observations.

Machine-learning algorithms can be classified along two main lines: **supervised** and **unsupervised** classification.

Supervised algorithms need a **training set**, where the objects are described in terms of attributes and belong to a class.

The performance of the resulting classifier is measured against a **test set**.

We can also use N-fold cross validation, where the test set is selected randomly from the training set N times, usually 10.

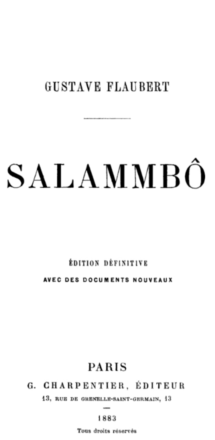
Unsupervised algorithms consider objects, where no class is provided.

Unsupervised algorithms learn regularities in data sets.

A Data Set: *Salammbô*

A corpus is a collection – a body – of texts.

French original	English translation



GUSTAVE FLAUBERT

SALAMMBÔ

ÉDITION DÉFINITIVE

AVEC DES DOCUMENTS NOUVEAUX

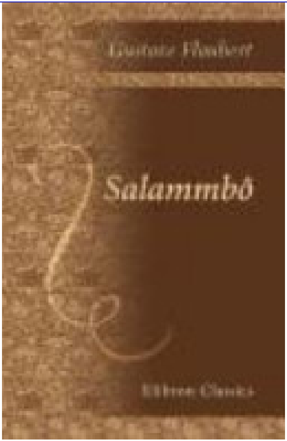
PARIS

G. CHARPENTIER, ÉDITEUR

13, RUE DE GRENELLE-SAINT-GERMAIN, 13

1883

Tous droits réservés



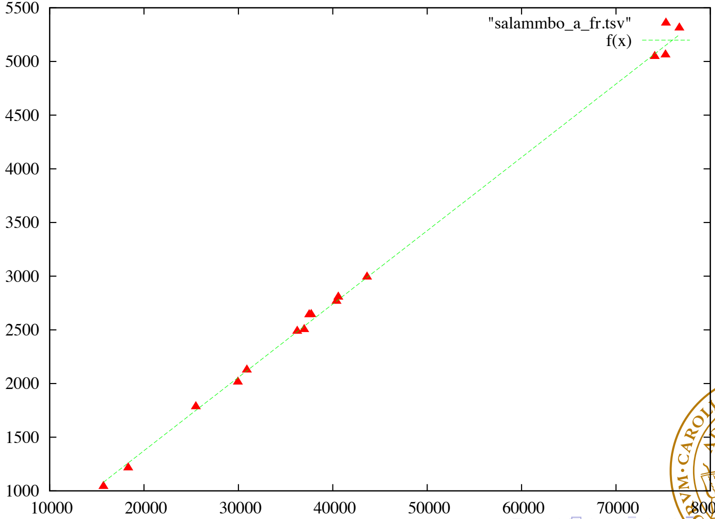
Supervised Learning

Letter counts from *Salammbô*

Chapter	French		English	
	# characters	# *A*	# characters	# *A*
Chapter 1	36,961	2,503	35,680	2,217
Chapter 2	43,621	2,992	42,514	2,761
Chapter 3	15,694	1,042	15,162	990
Chapter 4	36,231	2,487	35,298	2,274
Chapter 5	29,945	2,014	29,800	1,865
Chapter 6	40,588	2,805	40,255	2,606
Chapter 7	75,255	5,062	74,532	4,805
Chapter 8	37,709	2,643	37,464	2,396
Chapter 9	30,899	2,126	31,030	1,993
Chapter 10	25,486	1,784	24,843	1,627
Chapter 11	37,497	2,641	36,172	2,375
Chapter 12	40,398	2,766	39,552	2,560
Chapter 13	74,105	5,047	72,545	4,597
Chapter 14	76,725	5,312	75,352	4,871
Chapter 15	18,317	1,215	18,031	1,119

Supervised Learning: Regression

Letter count from *Salammbô* in French



Models

We will assume that data sets are governed by functions or models.
For instance given the set:

$$\{(\mathbf{x}_i, y_i) | 0 < i \leqslant N\},$$

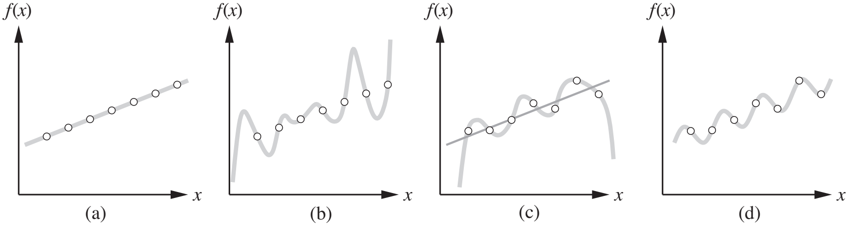
there exists a function such that:

$$f(\mathbf{x}_i) = y_i.$$

Supervised machine learning algorithms will produce hypothesized functions or models fitting the data.

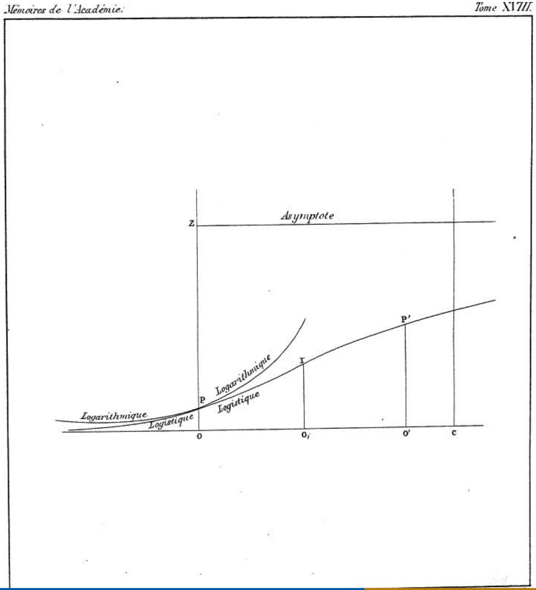
Selecting a Model

Often, multiple models can fit a data set.
The figure below shows four curves fitting the two data sets.

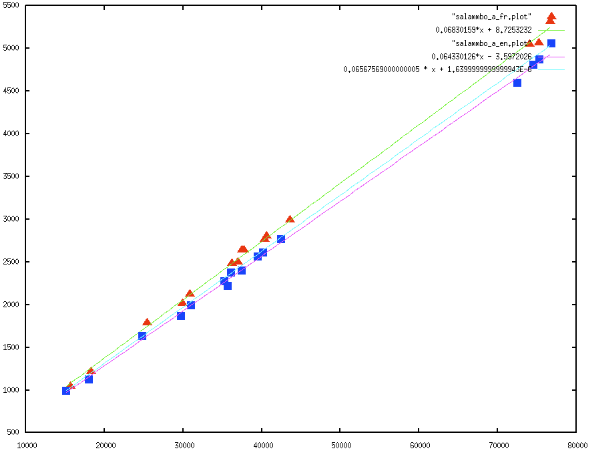


From the textbook, Stuart Russell and Peter Norvig, *Artificial Intelligence*, 3rd ed., 2010, page 696.
A general rule in machine learning is to prefer the simplest hypotheses, here the lower polynomial degrees.

Regression: Another Model, the Logistic Curve (Verhulst)



Supervised Learning: Classification



Given the data set, $\{(\mathbf{x}_i, y_i) | 0 < i \leqslant N\}$ and a model f, classification: $f(\mathbf{x}) = y$ is discrete, regression: $f(\mathbf{x}) = y$ is continuous.

Classification Data Set

Here a binary classification.

	# char.	# A	class (y)	# char.	# A	class (y)
Chapter 1	36,961	2,503	1	35,680	2,217	0
Chapter 2	43,621	2,992	1	42,514	2,761	0
Chapter 3	15,694	1,042	1	15,162	990	0
Chapter 4	36,231	2,487	1	35,298	2,274	0
Chapter 5	29,945	2,014	1	29,800	1,865	0
Chapter 6	40,588	2,805	1	40,255	2,606	0
Chapter 7	75,255	5,062	1	74,532	4,805	0
Chapter 8	37,709	2,643	1	37,464	2,396	0
Chapter 9	30,899	2,126	1	31,030	1,993	0
Chapter 10	25,486	1,784	1	24,843	1,627	0
Chapter 11	37,497	2,641	1	36,172	2,375	0
Chapter 12	40,398	2,766	1	39,552	2,560	0
Chapter 13	74,105	5,047	1	72,545	4,597	0
Chapter 14	76,725	5,312	1	75,352	4,871	0
Chapter 15	18,317	1,215	1	18,031	1,119	0

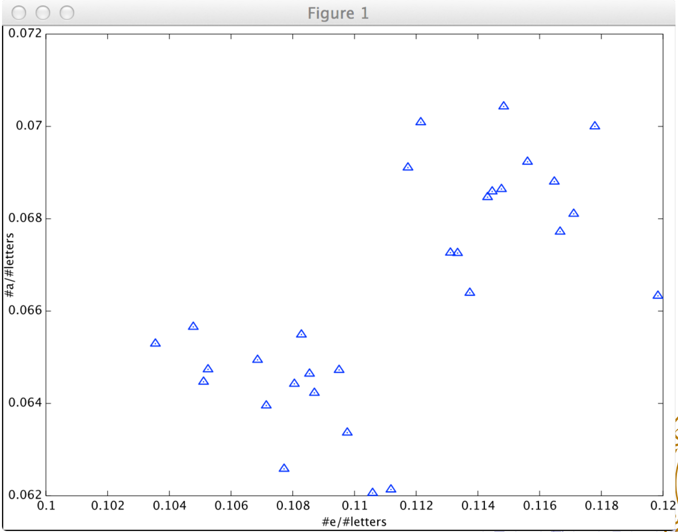
Supervised Learning: Fisher's Iris data set (1936)

Table I

Iris setosa				*Iris versicolor*				*Iris virginica*			
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5·1	3·5	1·4	0·2	7·0	3·2	4·7	1·4	6·3	3·3	6·0	2·5
4·9	3·0	1·4	0·2	6·4	3·2	4·5	1·5	5·8	2·7	5·1	1·9
4·7	3·2	1·3	0·2	6·9	3·1	4·9	1·5	7·1	3·0	5·9	2·1
4·6	3·1	1·5	0·2	5·5	2·3	4·0	1·3	6·3	2·9	5·6	1·8
5·0	3·6	1·4	0·2	6·5	2·8	4·6	1·5	6·5	3·0	5·8	2·2
5·4	3·9	1·7	0·4	5·7	2·8	4·5	1·3	7·6	3·0	6·6	2·1
4·6	3·4	1·4	0·3	6·3	3·3	4·7	1·6	4·9	2·5	4·5	1·7
5·0	3·4	1·5	0·2	4·9	2·4	3·3	1·0	7·3	2·9	6·3	1·8
4·4	2·9	1·4	0·2	6·6	2·9	4·6	1·3	6·7	2·5	5·8	1·8
4·9	3·1	1·5	0·1	5·2	2·7	3·9	1·4	7·2	3·6	6·1	2·5
5·4	3·7	1·5	0·2	5·0	2·0	3·5	1·0	6·5	3·2	5·1	2·0
4·8	3·4	1·6	0·2	5·9	3·0	4·2	1·5	6·4	2·7	5·3	1·9
4·8	3·0	1·4	0·1	6·0	2·2	4·0	1·0	6·8	3·0	5·5	2·1
4·3	3·0	1·1	0·1	6·1	2·9	4·7	1·4	5·7	2·5	5·0	2·0
5·8	4·0	1·2	0·2	5·6	2·9	3·6	1·3	5·8	2·8	5·1	2·4
5·7	4·4	1·5	0·4	6·7	3·1	4·4	1·4	6·4	3·2	5·3	2·3
5·4	3·9	1·3	0·4	5·6	3·0	4·5	1·5	6·5	3·0	5·5	1·8
5·1	3·5	1·4	0·3	5·8	2·7	4·1	1·0	7·7	3·8	6·7	2·2
5·7	3·8	1·7	0·3	6·2	2·2	4·5	1·5	7·7	2·6	6·9	2·3
5·1	3·8	1·5	0·3	5·6	2·5	3·9	1·1	6·0	2·2	5·0	1·5
5·4	3·4	1·7	0·2	5·9	3·2	4·8	1·8	6·9	3·2	5·7	2·3
5·1	3·7	1·5	0·4	6·1	2·8	4·0	1·3	5·6	2·8	4·9	2·0

Unsupervised Learning: Clustering

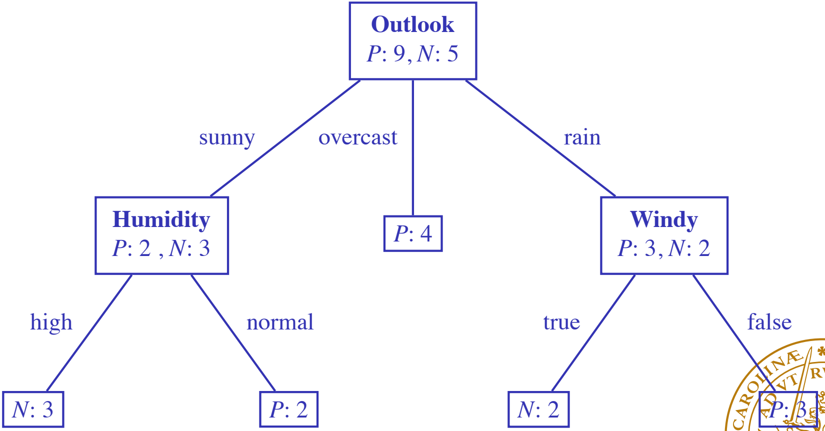
No class is given:



Objects, Attributes, and Classes. After Quinlan (1986)

Object	Attributes				Class
	Outlook	**Temperature**	**Humidity**	**Windy**	
1	Sunny	Hot	High	False	*N*
2	Sunny	Hot	High	True	*N*
3	Overcast	Hot	High	False	*P*
4	Rain	Mild	High	False	*P*
5	Rain	Cool	Normal	False	*P*
6	Rain	Cool	Normal	True	*N*
7	Overcast	Cool	Normal	True	*P*
8	Sunny	Mild	High	False	*N*
9	Sunny	Cool	Normal	False	*P*
10	Rain	Mild	Normal	False	*P*
11	Sunny	Mild	Normal	True	*P*
12	Overcast	Mild	High	True	*P*
13	Overcast	Hot	Normal	False	*P*
14	Rain	Mild	High	True	*N*

Classifying Objects with Decision Trees. After Quinlan (1986)



Decision Trees and Classification

Each object is defined by an attribute vector (or feature vector) $\{A_1, A_2, ..., A_v\}$
Each object belongs to one class $\{C_1, C_2, ..., C_n\}$
The attributes of the examples are:
$\{Outlook, Temperature, Humidity, Windy\}$ and the classes are: $\{N, P\}$.
The nodes of the tree are the attributes.
Each attribute has a set of possible values. The values of *Outlook* are $\{sunny, rain, overcast\}$
The branches correspond to the values of each attribute
The optimal tree corresponds to a minimal number of tests.

Entropy, Decision Trees, and Classification

Decision trees are useful devices to classify objects into a set of classes.
Entropy can help us learn automatically decision trees from a set of data.
The algorithm is one of the simplest machine-learning techniques to obtain a classifier.

Inducing (Learning) Decision Trees Automatically: ID3

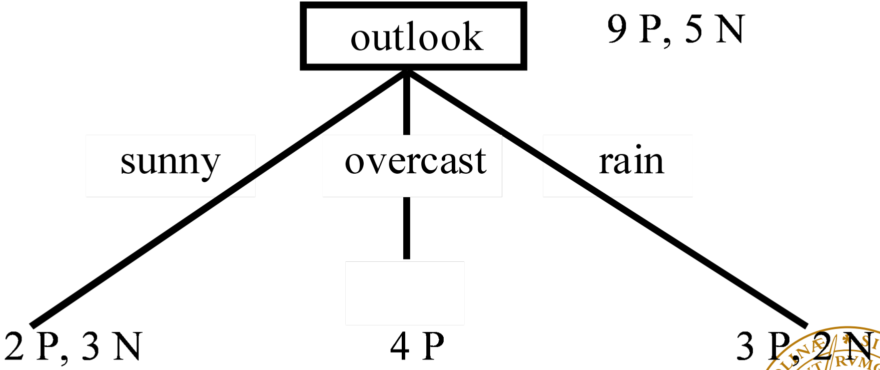
It is possible to design many trees that classify the objects successfully
An efficient decision tree uses a minimal number of tests.
In the decision tree:

- Each example is defined by a finite number of attributes.
- Each node in the decision tree corresponds to an attribute that has as many branches as the attribute has possible values.

At the root of the tree, the condition must be the most discriminating, that is, have branches gathering most positive examples while others gather negative examples.

ID3 (Quinlan 1986)

Each attribute scatters the set into as many subsets as there are values for this attribute.



At each decision point, the "best" attribute has the maximal separation power, the maximal information gain

Entropy

Information theory models a text as a sequence of symbols.
Let $x_1, x_2, \ldots, x_N$ be a discrete set of N symbols representing the characters.
The information content of a symbol is defined as

$$I(x_i) = -\log_2 p(x_i) = \log_2 \frac{1}{p(x_i)},$$

where

$$p(x_i) = \frac{Count(x_i)}{\sum_{j=1}^{n} Count(x_j)}.$$

Entropy, the average information content, is defined as:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x),$$

By convention: $0 \log_2 0 = 0$.

Entropy of a Text

The entropy of the text is

$$\begin{aligned} H(X) &= -\sum_{x \in X} p(x) \log_2 p(x). \\ &= -p(A)\log_2 p(A) - p(B)\log_2 p(B) - \ldots \\ &\quad -p(Z)\log_2 p(Z) - p(\grave{A})\log_2 p(\grave{A}) - \ldots \\ &\quad -p(\ddot{Y})\log_2 p(\ddot{Y}) - p(blanks)\log_2 p(blanks). \end{aligned}$$

Entropy of Gustave Flaubert's *Salammbô* in French is $H(X) = 4.39$.

Cross-Entropy

The cross entropy of m on p is defined as:

$$H(p,m) = -\sum_{x \in X} p(x) \log_2 m(x).$$

We have the inequality $H(p) \leq H(p,m)$.

	Entropy	Cross entropy	Difference
Salammbô, chapters 1-14, training set	4.39481	4.39481	0.0
Salammbô, chapter 15, test set	4.34937	4.36074	0.01137
Notre Dame de Paris, test set	4.43696	4.45507	0.01811
Nineteen Eighty-Four, test set	4.35922	4.82012	0.46090

ID3 (Quinlan 1986)

ID3 uses the entropy to select the best attribute to be the root of the tree and recursively the next attributes of the resulting nodes.
The entropy of a two-class set p and n is:

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}.$$

The weighted average of all the nodes below an attribute A is:

$$E(A) = \sum_{i=1}^{v}\frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}).$$

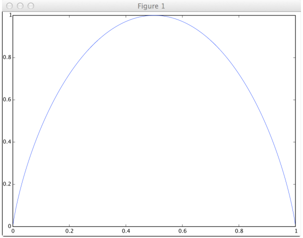
The information gain is defined as $I(p,n) - E(A)$ (or $I_{before} - I_{after}$)
This measures the separating power of an attribute: the more the gain, the better the attribute.

Understanding the Entropy

For a binary classification, we set:

$$x = \frac{p}{p+n} \text{ and } \frac{n}{p+n} = 1 - x.$$

$$I(x) = -x \log_2 x - (1-x)\log_2(1-x) \text{ with } x \in [0,1].$$



The entropy reaches a maximum when there are as many positive as negative examples in the data set. It is minimal when the set consists of either positive or negative examples.

Example

The entropy of the data set is:

$$I(p, n) = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940.$$

Outlook has three values: *sunny*, *overcast*, and *rain*. The entropies of the corresponding subsets are:

$$\begin{array}{llll} sunny: & I(p_1, n_1) & = & -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971. \\ overcast: & I(p_2, n_2) & = & 0.0 \\ rain: & I(p_3, n_3) & = & -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971. \end{array}$$

$Gain(Outlook)$ is then $0.940 - \frac{5}{14} \times 0.971 - \frac{5}{14} \times 0.971 = 0.246$, the highest possible among the attributes.
We have $Gain(Temperature) = 0.029$, $Gain(Humidity) = 0.151$, and $Gain(Windy) = 0.048$.

Inducing Decision Trees Automatically: ID3

The algorithm to build the decision tree is simple.

The information gain is computed on the data set for all attributes in the set of attributes, SA.

The attribute with the highest gain is selected to be the root of the tree:

$$A = \arg\max_{a \in SA} I(n, p) - E(a).$$

The data set is split into v subsets $\{N_1, ..., N_v\}$, where the value of A for the objects in N_i is A_i

For each subset, a corresponding node is created below the root.

This process is repeated recursively for each node of the tree with the subset it contains until all the objects of the node are either positive or negative.

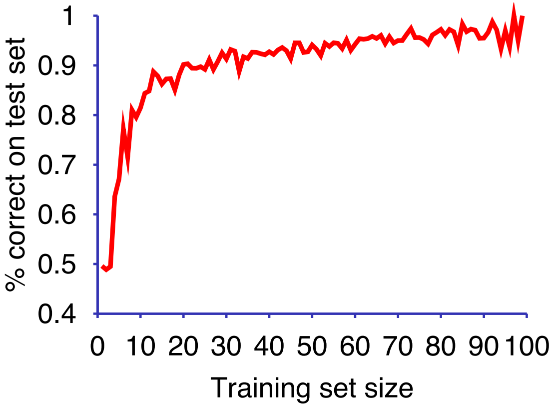
For a training set of N instances each having M attributes, ID3's complexity to generate a decision tree is $O(NM)$.

Algorithm (From the Textbook)

function DTL(*examples*, *attributes*, *parent_examples*) **returns** a tree
  if *examples* is empty **then**
    return PLURALITYVALUE(*parent_examples*)
  else if all *examples* have the same classification **then**
    return the classification
  else if *attributes* is empty **then**
    return PLURALITYVALUE(*examples*)
  else
    $A \leftarrow \arg\max_{a \in attributes} \textsc{InformationGain}(a, examples)$
    tree ← a new decision tree with root test A
    for all $v_k \in A$ **do**
      $exs \leftarrow \{e : e \in examples$ **and** $e.A = v_k\}$
      $subtree \leftarrow \mathrm{DTL}(exs, attributes - A, examples)$
      add a branch to *tree* with label $\{A = v_k\}$ and subtree *subtree*
    return *tree*

Learning Curve

The classical evaluation technique uses a training set and a test set.
Generally, the larger the training set, the better the performance.
This can be visualized with a learning curve. From the textbook, Stuart Russell and Peter Norvig, *Artificial Intelligence*, 3rd ed., 2010, page 703.



Overfitting

When two classifiers have equal performances on a specific test set, the simplest one is supposed to be more general

A small decision tree is always preferable to a larger one.

Complex classifiers may show an overfit to the training data and have poor performance when the data set changes.

In the case of decision trees, overfits show with deep trees and when numerous leaves have few examples.

Pruning Trees

Decision tree pruning consists in removing and merging leaves with few objects, for instance one or two.

To prune the tree:

1. Generate the complete decision tree
2. Consider the nodes that have only leaves as children and merge the leaves if the information gain is below a certain threshold.
3. To determine the threshold, we use a significance test.

This is the χ^2 pruning.

Contingency Tables

Example: A set of 12 positive and 8 negative examples.
Let us compare two attributes:

- Attribute 1 with three values: $\{v_1^1, v_1^2, v_1^3\}$
- Attribute 2 with three values: $\{v_2^1, v_2^2, v_2^3\}$

Contingency tables are devices to visualize frequency distributions.

Attribute 1		v_1^1	v_1^2	v_1^3	
	Positive	6	3	3	12 (60%)
	Negative	4	2	2	8 (40%)
	Total	10	5	5	20 (100%)

Attribute 2		v_1^1	v_1^2	v_1^3	
	Positive	1	1	10	12 (60%)
	Negative	8	0	0	8 (40%)
	Total	9	1	10	20 (100%)

Significance Test

An attribute with no significance would have the same proportions of positive and negative examples before and after the test.

Proportions before, for instance 12 and 8:

$$\frac{p}{p+n} \quad \text{and} \quad \frac{n}{p+n}$$

Proportions after, for v_k, a value of the attribute:

$$\frac{p_k}{p_k+n_k} \quad \text{and} \quad \frac{n_k}{p_k+n_k}$$

The expected values of a nonsignificant attribute (null hypothesis) are:

$$\hat{p}_k = \frac{p}{p+n} \times (p_k + n_k) \qquad \hat{n}_k = \frac{n}{p+n} \times (p_k + n_k)$$

The total deviation for an attribute with values $v_1..v_d$ is:

$$\Delta = \sum_{k=1}^{d} \frac{(p_k - \hat{p}_k)^2}{\hat{p}_k} + \frac{(n_k - \hat{n}_k)^2}{\hat{n}_k}$$

Pruning Trees (II)

The null hypothesis corresponds to an irrelevant attribute: $\Delta = 0$.
Δ values are distributed according to the χ^2 distribution
If v is the number of values of an attribute, the degree of freedom is $v - 1$
The corresponding χ^2 distributions can be found in statistical tables.
For three degrees of freedom:
10%: 6.25, 5%: 7.81, 1%: 11.34, 1‰: 16.27
The significance level is usually 5%
An alternate pruning method is to modify the termination condition of the algorithm:
We stop when the information gain is below a certain threshold.

Dealing with Real Data

- Unknown attribute values: Use the most frequent value of the attribute or all the values;
- Attributes with many values (large attribute domains): Difficult to handle for decision trees. The book suggests to use the gain ratio:

$$\frac{\text{Information gain}}{\text{Information value}}, \text{where Information value} = -\sum_{i=1}^{v} \frac{p_i + n_i}{p+n} \log_2 \frac{p_i + n_i}{p+n}$$

- Numerical attributes: Find the binary split point that maximizes the information gain
- Numerical output: Regression tree

Evaluation

The standard evaluation procedure is to train the classifier on a training set and evaluate the performance on a test set.

When we have only one set, we divide it in two subsets: the training set and the test set (or holdout data).

The split can be 90–10 or 80–20

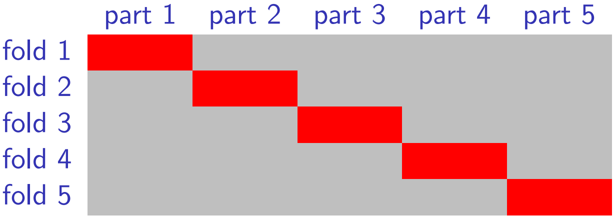
This often optimizes the classifier for a specific test set and creates an overfit

Cross Validation

A N-fold cross validation mitigates the overfit

The set is partitioned into N subsets, $N = 5$ for example, one of them being the test set (red) and the rest the training set (gray).

The process is repeated N times with a different test set: N folds



At the extreme, leave-one-out cross-validation

Model Selection

Validation can apply to one classification method

We can use it to select a classification method and its parametrization.

Needs three sets: training set, development set, and test set.

Loss

Depending on their type, errors can have different impacts.

Imagine a smoke detector that predicts fire.

Equivalent to a classifier that uses input data from sensors and classifies them as *fire* and *nonfire*.

The loss function is defined as $L(y, \hat{y})$ with $\hat{y} = h(\mathbf{x})$, where $\mathbf{x}$ is the vector of attributes, h the classifier, and y, the correct value.

The cost of $L(\mathit{fire}, \mathit{nonfire})$ is much higher than $L(\mathit{nonfire}, \mathit{fire})$.

$L(\mathit{fire}, \mathit{fire}) = L(\mathit{nonfire}, \mathit{nonfire}) = 0$

Common Loss Measures

$$\begin{array}{lcll} L_1(y, \hat{y}) & = & |y - \hat{y}| & \text{Absolute value loss} \\ L_2(y, \hat{y}) & = & (y - \hat{y})^2 & \text{Squared error loss} \\ L_{0/1}(y, \hat{y}) & = & 0 \text{ if } y = \hat{y} \text{ else } 1 & \text{0/1 loss} \end{array}$$

Empirical Loss

We compute the empirical loss of a classifier h on a set of examples E using the formula:

$$Loss(L, E, h) = \frac{1}{N} \sum_{E} L(y, h(x)).$$

For continuous functions:

$$Loss(L, E, h) = \frac{1}{N} \sum_{E} (y - h(x))^2.$$

Evaluation

There are different kinds of measures to evaluate the performance of machine learning techniques, for instance:

- Precision and recall in information retrieval and natural language processing;
- The *receiver operating characteristic* (ROC) in medicine.

	Positive examples: P	**Negative examples:** N
Classified as P	True positives: A	False positives: B
Classified as N	False negatives: C	True negatives: D

More on the receiver operating characteristic here: `http://en.wikipedia.org/wiki/Receiver_operating_characteristic`

Recall, Precision, and the F-Measure

The **accuracy** is $\dfrac{|A \cup D|}{|P \cup N|}$.

Recall measures how much relevant examples the system has classified correctly, for P:

$$\text{Recall} = \frac{|A|}{|A \cup C|}.$$

Precision is the accuracy of what has been returned, for P:

$$\text{Precision} = \frac{|A|}{|A \cup B|}.$$

Recall and precision are combined into the **F-measure**, which is defined as the harmonic mean of both numbers:

$$F = \frac{2 \cdot \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

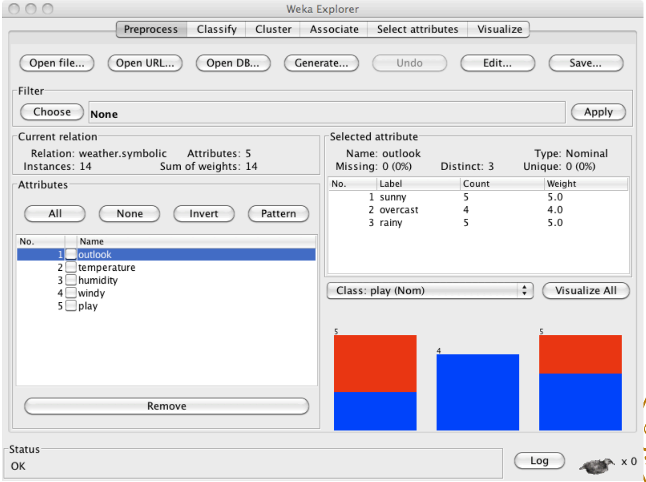
Measuring Quality: The Confusion Matrix

From Franz (1996, p. 124)

↓Correct	Tagger →									
	DT	IN	JJ	NN	RB	RP	VB	VBD	VBG	VBN
DT	99.4	0.3	–	–	0.3	–	–	–	–	–
IN	0.4	97.5	–	–	1.5	0.5	–	–	–	–
JJ	–	0.1	93.9	1.8	0.9	–	0.1	0.1	0.4	1.5
NN	–	–	2.2	95.5	–	–	0.2	–	0.4	–
RB	0.2	2.4	2.2	0.6	93.2	1.2	–	–	–	–
RP	–	24.7	–	1.1	12.6	61.5	–	–	–	–
VB	–	–	0.3	1.4	–	–	96.0	–	–	0.2
VBD	–	–	0.3	–	–	–	–	94.6	–	4.8
VBG	–	–	2.5	4.4	–	–	–	–	93.0	–
VBN	–	–	4.6	–	–	–	–	4.3	–	90.6

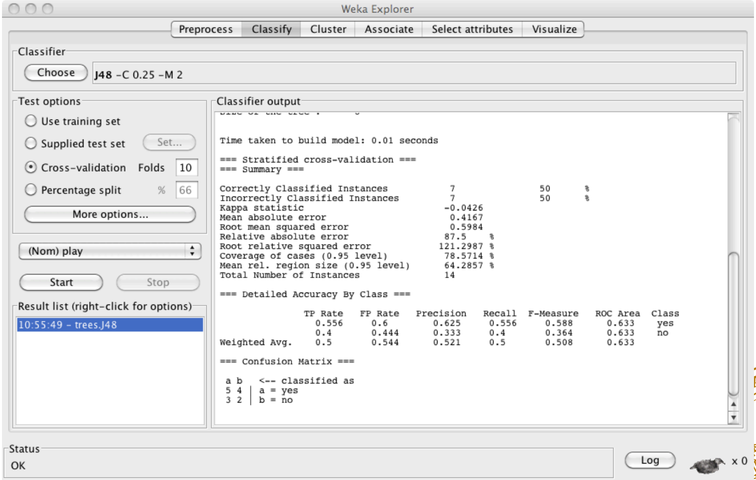
The Weka Toolkit

Weka: A powerful collection of machine-learning algorithms
`http://www.cs.waikato.ac.nz/ml/weka/`.



The Weka Toolkit

Running ID3



ARFF: The Weka Data Format

Storing Quinlan's data set in Weka's attribute-relation file format (ARFF) http://weka.wikispaces.com/ARFF:

```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
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
```