

Language Technology

<http://cs.lth.se/edan20/>

Chapter 4: Topics in Information Theory and Machine Learning

Pierre Nugues

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

September 5, 2022



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

- 1 Machine learning always starts with **data sets**: a collection of objects or observations.
- 2 Machine-learning algorithms can be classified along two main lines: **supervised** and **unsupervised** classification.
- 3 Supervised algorithms need a **training set**, where the objects are described in terms of attributes and belong to a known class or have a known output.
- 4 The performance of the resulting classifier is measured against a **test set**.
- 5 We can also use N -fold cross validation, where the test set is selected randomly from the training set N times, usually 10.
- 6 Unsupervised algorithms consider objects, where no class is provided.
- 7 Unsupervised algorithms learn regularities in data sets.

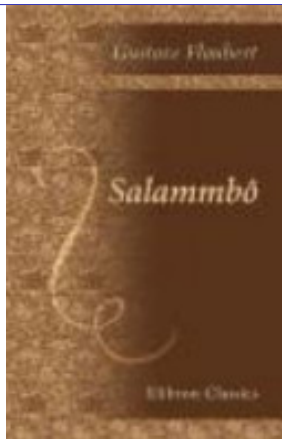
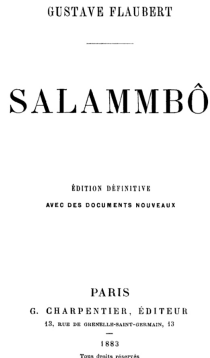


A Dataset: *Salammbô*

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

French original

English translation



Supervised Learning

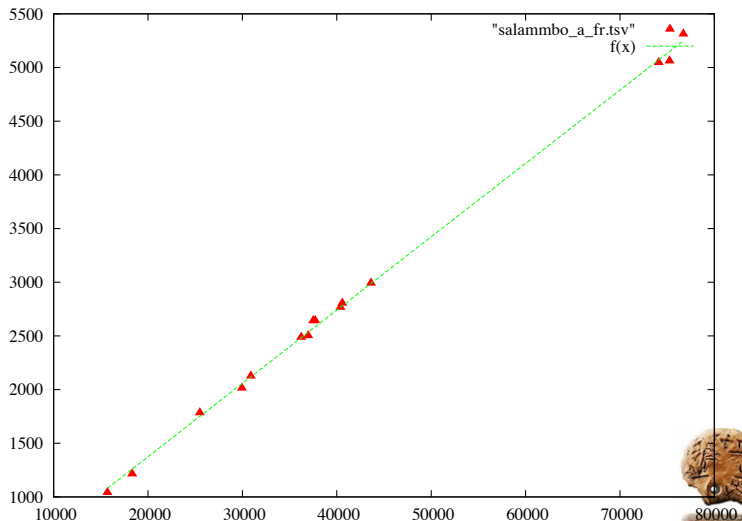
Letter counts from *Salammbo*

| 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:

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

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.



Notations

We will follow these notations:

- \mathbf{x} , the vector representing an observation (or sample, or example, or input);
in *Salammbo*, an observation is the number of letters in a chapter.
We have 15 observations;
- y , the observed response (or target, or output); in programs, the variable names are `y` or `y_true`;
in *Salammbo*, the number of As in a chapter. We have 15 responses;
- \hat{y} , the value predicted by the model; in programs, the variable names are `y_pred` or `y_hat`;
- \mathbf{w} , the weights or parameters of the model, so that $\mathbf{w} \cdot \mathbf{x} = \hat{y}$;
another possible notation for \mathbf{w} is β
- \mathbf{X} , the matrix of all the observations
- \mathbf{y} , the vector of all the responses and $\hat{\mathbf{y}}$, for all the predictions



Selecting a Model

Often, multiple models can fit a data set:

Three polynomials of degree: 1, a straight line, 8, and 9 to fit the *Salammbô* dataset.



A general rule in machine learning is to prefer the simplest hypotheses, here the lower polynomial degrees. Otherwise, the model can **overfit** the data.

In our case, the optimal model \mathbf{w} has two parameters: (w_0, w_1) .



Loss or Objective Function

What are the optimal values of \mathbf{w} ?

The model should minimize the difference between:

- the predicted values $\hat{\mathbf{y}}$ and
- the observed values \mathbf{y} .

This is called the **loss**

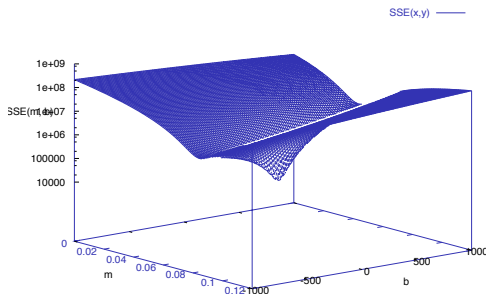
For *Salambô*, the loss is the *mean of the squared errors* (MSE):

$$\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$



Visualizing the Loss

$$\hat{y} = mx + b$$



We will use the notation

$$\hat{y} = w_1 x + w_0$$

to generalize to any dimension



The Matrices

$$\mathbf{X} = \begin{bmatrix} 1 & 36961 \\ 1 & 43621 \\ 1 & 15694 \\ 1 & 36231 \\ 1 & 29945 \\ 1 & 40588 \\ 1 & 75255 \\ 1 & 37709 \\ 1 & 30899 \\ 1 & 25486 \\ 1 & 37497 \\ 1 & 40398 \\ 1 & 74105 \\ 1 & 76725 \\ 1 & 18317 \end{bmatrix} ; \mathbf{w} = \begin{bmatrix} 8.7253 \\ 0.0683 \end{bmatrix} ; \mathbf{y} = \begin{bmatrix} 2533.22 \\ 2988.11 \\ 1080.65 \\ 2483.36 \\ 2054.02 \\ 2780.95 \\ 5148.76 \\ 2584.31 \\ 2119.18 \\ 1749.46 \\ 2569.83 \\ 2767.97 \\ 5070.21 \\ 5249.16 \\ 1259.81 \end{bmatrix} ; \mathbf{y} = \begin{bmatrix} 2503 \\ 2992 \\ 1042 \\ 2487 \\ 2014 \\ 2805 \\ 5062 \\ 2643 \\ 2126 \\ 1784 \\ 2641 \\ 2766 \\ 5047 \\ 5312 \\ 1215 \end{bmatrix} ; \mathbf{se} = \begin{bmatrix} 913.26 \\ 15.14 \\ 1493.86 \\ 13.25 \\ 1601.31 \\ 578.40 \\ 7527.51 \\ 3444.53 \\ 46.57 \\ 1193.04 \\ 5065.18 \\ 38920 \\ 538.909 \\ 3948.29 \\ 2603.53 \end{bmatrix} .$$



Minimizing the Loss

The loss function is convex and has a unique minimum.

The loss reaches a minimum when the partial derivatives are zero:

$$\begin{aligned}\frac{\partial \text{Loss}}{\partial m} &= \sum_{i=1}^q \frac{\partial}{\partial m} (y_i - (mx_i + b))^2 = -2 \sum_{i=1}^q x_i (y_i - (mx_i + b)) = 0 \\ \frac{\partial \text{Loss}}{\partial b} &= \sum_{i=1}^q \frac{\partial}{\partial b} (y_i - (mx_i + b))^2 = -2 \sum_{i=1}^q (y_i - (mx_i + b)) = 0\end{aligned}$$



The Gradient Descent

The gradient descent is a numerical method to find the minimum of $f(x_0, x_1, x_2, \dots, x_n) = y$, when there is no analytical solution.

Let us denote $\mathbf{x} = (x_0, x_1, x_2, \dots, x_n)$

We derive successive approximations to find the minimum of f :

$$f(\mathbf{x}_1) > f(\mathbf{x}_2) > \dots > f(\mathbf{x}_k) > f(\mathbf{x}_{k+1}) > \dots > \min$$

Points in the neighborhood of \mathbf{x} are defined by $\mathbf{x} + \mathbf{v}$ with $\|\mathbf{v}\|$ small

Given \mathbf{x} , find \mathbf{v} subject to $f(\mathbf{x}) > f(\mathbf{x} + \mathbf{v})$



The Gradient Descent (Cauchy, 1847)

Using a Taylor expansion: $f(\mathbf{x} + \mathbf{v}) = f(\mathbf{x}) + \mathbf{v} \cdot \nabla f(\mathbf{x}) + \dots$

The gradient is a direction vector corresponding to the steepest slope:

$$\nabla f(x_0, x_1, x_2, \dots, x_n) = \left(\frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right).$$

$f(\mathbf{x} + \mathbf{v})$ reaches a minimum or a maximum when \mathbf{v} and $\nabla f(\mathbf{x})$ are colinear:

- Steepest ascent: $\mathbf{v} = \alpha \nabla f(\mathbf{x})$,
- Steepest descent: $\mathbf{v} = -\alpha \nabla f(\mathbf{x})$,

where $\alpha > 0$.

We have then: $f(\mathbf{x} - \alpha \nabla f(\mathbf{x})) \approx f(\mathbf{x}) - \alpha \|\nabla f(\mathbf{x})\|^2$.

The inequality:

$$f(\mathbf{x}) > f(\mathbf{x} - \alpha \nabla f(\mathbf{x}))$$

enables us to move one step down to the minimum.

We then use the iteration:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k).$$



Classification Dataset

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 |



Separating Classes



Given the data set, $\{(\mathbf{x}_i, y_i) | 0 < i \leq N\}$ and a model f :

- Classification: $f(\mathbf{x}) = y$ is discrete,
- Regression: $f(\mathbf{x}) = y$ is continuous.



Supervised Machine-Learning Algorithms

Linear classifiers:

- 1 Perceptron
- 2 Logistic regression
- 3 Neural networks (with many flavors)



Classification

We represent classification using a threshold function (a variant of the signum function):

$$H(\mathbf{w} \cdot \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

The classification function associates P with 1 and N with 0.
We want to find the separating hyperplane:

$$\begin{aligned} \hat{y}(\mathbf{x}) &= H(\mathbf{w} \cdot \mathbf{x}) \\ &= H(w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n), \end{aligned}$$

given a data set of q examples: $DS = \{(1, x_1^j, x_2^j, \dots, x_n^j, y^j) | j : 1..q\}$.

We use $x_0 = 1$ to simplify the equations.

For a binary classifier, y has then two possible values $\{0, 1\}$ corresponding in our example to $\{\text{French, English}\}$.



Berkson's Data Set (1944)

| Drug concentration | Number exposed | Survive Class 0 | Die Class 1 | Mortality rate | Expected mortality |
|--------------------|----------------|-----------------|-------------|----------------|--------------------|
| 40 | 462 | 352 | 110 | .2359 | .2206 |
| 60 | 500 | 301 | 199 | .3980 | .4339 |
| 80 | 467 | 169 | 298 | .6380 | .6085 |
| 100 | 515 | 145 | 370 | .7184 | .7291 |
| 120 | 561 | 102 | 459 | .8182 | .8081 |
| 140 | 469 | 69 | 400 | .8529 | .8601 |
| 160 | 550 | 55 | 495 | .9000 | .8952 |
| 180 | 542 | 43 | 499 | .9207 | .9195 |
| 200 | 479 | 29 | 450 | .9395 | .9366 |
| 250 | 497 | 21 | 476 | .9577 | .9624 |
| 300 | 453 | 11 | 442 | .9757 | .9756 |

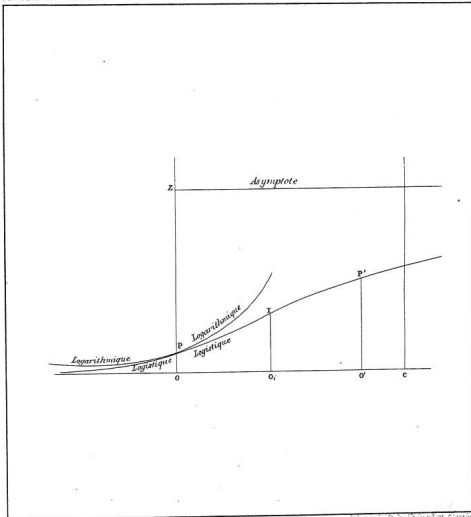
Table: A data set. Adapted and simplified from the original article that described how to apply logistic regression to classification by Joseph Berkson, Application of the Logistic Function to Bio-Assay. *Journal of the American Statistical Association* (1944).



Another Model, the Logistic Curve (Verhulst)

Mémoires de l'Académie.

Tome XVIII.



Mémoire sur la population par M. P. Verhulst.

$$\text{Logistic}(x) = \frac{1}{1 + e^{-x}}$$

$$\begin{aligned} \hat{y}(\mathbf{x}) &= \text{Logistic}(\mathbf{w} \cdot \mathbf{x}) \\ &= \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}} \end{aligned}$$



Loss: Binary Cross Entropy

In practice, we use the mean and the natural logarithm:

$$H(P, M) = -\frac{1}{|X|} \sum_{x \in X} P(x) \log M(x),$$

where P is the truth, and M is the prediction of the model, a probability in the case of logistic regression.

In binary classification:

- $P(x) = 1$
- $M(x)$ is the predicted probability of being class 1.
- If the observation belongs to class 0, its predicted probability is $1 - M(x)$.



Example of Cross Entropy

Computing the cross-entropy of six observations:

| Observations | 1 | 2 | 3 | 4 | 5 | 6 |
|--------------------------------------|--------|--------|--------|--------|--------|--------|
| Dose | 140 | 300 | 140 | 160 | 140 | 250 |
| Observed class (Truth) | 0 | 1 | 1 | 1 | 1 | 1 |
| Model prediction of being class 1 | 0.3487 | 0.9964 | 0.8557 | 0.9056 | 0.8557 | 0.9882 |
| Model prediction of being class 0 | 0.6513 | | | | | |
| $-P(x)\log M(x)$: | 0.4287 | 0.0036 | 0.1559 | 0.0992 | 0.1559 | 0.0119 |

Mean = 0.14252826



Code Example: Logistic Regression with sklearn

Experiment: Jupyter Notebook:

https://github.com/pnugues/ilppp/blob/master/programs/ch04/sklearn/salamambo_dataset.ipynb



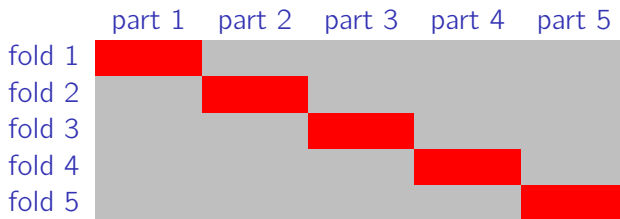
Evaluation

- 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



Types of Iris



Iris virginica



Iris setosa



Iris versicolor

Courtesy Wikipedia



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

180 MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS

Table I

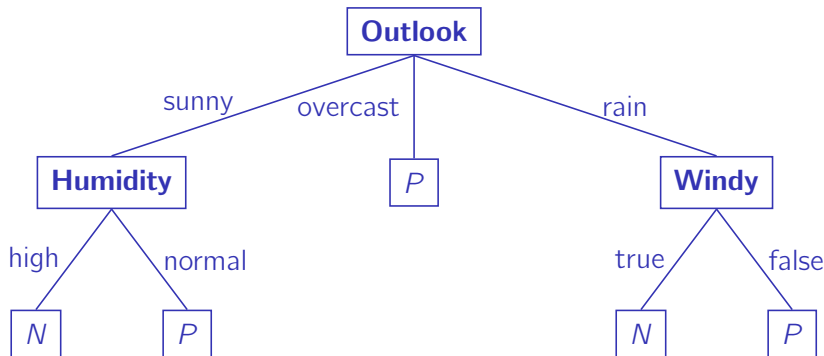
| <i>Iris setosa</i> | | | | <i>Iris versicolor</i> | | | | <i>Iris virginica</i> | | | |
|--------------------|-------------|--------------|-------------|------------------------|-------------|--------------|-------------|-----------------------|-------------|--------------|-------------|
| Sepal length | 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 |

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)



Matrix Notation

- A feature vector (predictors): \mathbf{x} , and feature matrix: \mathbf{X} ;
- The class: y and the class vector: \mathbf{y} ;
- The predicted class (response): \hat{y} , and predicted class vector: $\hat{\mathbf{y}}$

$$\mathbf{X} = \begin{bmatrix} \text{Sunny} & \text{Hot} & \text{High} & \text{False} \\ \text{Sunny} & \text{Hot} & \text{High} & \text{True} \\ \text{Overcast} & \text{Hot} & \text{High} & \text{False} \\ \text{Rain} & \text{Mild} & \text{High} & \text{False} \\ \text{Rain} & \text{Cool} & \text{Normal} & \text{False} \\ \text{Rain} & \text{Cool} & \text{Normal} & \text{True} \\ \text{Overcast} & \text{Cool} & \text{Normal} & \text{True} \\ \text{Sunny} & \text{Mild} & \text{High} & \text{False} \\ \text{Sunny} & \text{Cool} & \text{Normal} & \text{False} \\ \text{Rain} & \text{Mild} & \text{Normal} & \text{False} \\ \text{Sunny} & \text{Mild} & \text{Normal} & \text{True} \\ \text{Overcast} & \text{Mild} & \text{High} & \text{True} \\ \text{Overcast} & \text{Hot} & \text{Normal} & \text{False} \\ \text{Rain} & \text{Mild} & \text{High} & \text{True} \end{bmatrix}; \mathbf{y} = \begin{bmatrix} \text{N} \\ \text{N} \\ \text{P} \\ \text{P} \\ \text{P} \\ \text{N} \\ \text{P} \\ \text{N} \\ \text{P} \\ \text{P} \\ \text{P} \\ \text{P} \\ \text{P} \\ \text{N} \end{bmatrix}$$



Converting Symbolic Attributes into Numerical Vectors

Linear classifiers are numerical systems.

Symbolic – nominal – attributes are mapped onto vectors of binary values.

A conversion of the weather data set.

| Object | Attributes | | | | | | | | | | Class |
|--------|------------|----------|------|-------------|------|------|----------|--------|-------|-------|-------|
| | Outlook | | | Temperature | | | Humidity | | Windy | | |
| | Sunny | Overcast | Rain | Hot | Mild | Cool | High | Normal | True | False | |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | N |
| 2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | N |
| 3 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | P |
| 4 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | P |
| 5 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | P |
| 6 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | N |
| 7 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | P |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | N |
| 9 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | P |
| 10 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | P |
| 11 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | P |
| 12 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | P |
| 13 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | P |
| 14 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | N |



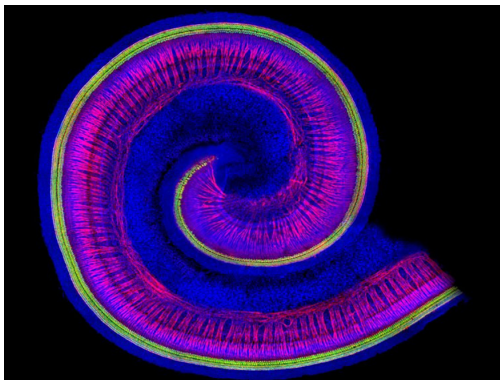
Code Example: Categorical Data with sklearn

Experiment: Jupyter Notebook:

https://github.com/pnugues/ilppp/blob/master/programs/ch04/sklearn/quinlan_dataset.ipynb



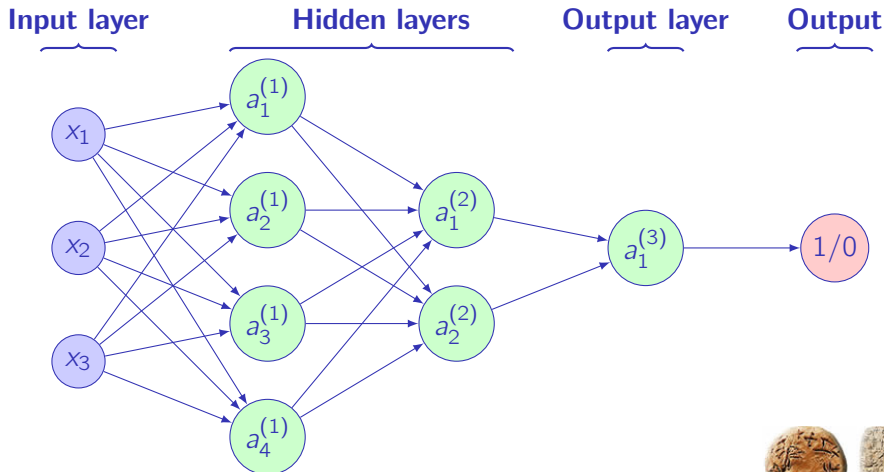
Neural Networks



A photomicrograph showing the classic view of the snail-shaped cochlea with hair cells stained green and neurons showing reddish-purple. [Decibel Therapeutics (<https://www.decibeltx.com>)]. Source: <https://www.genengnews.com/insights/targeting-the-inner-ear/>

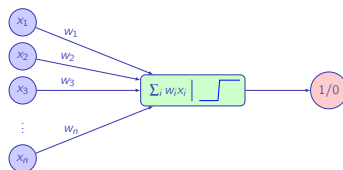


Neural Networks: Computer Model

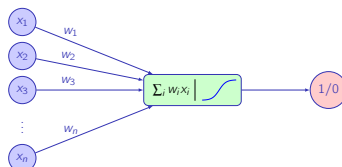


Activation Functions

Heaviside (perceptron)



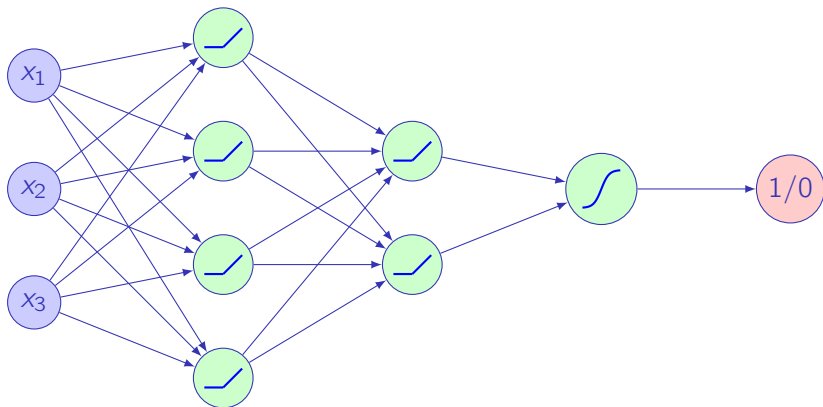
Logistic function (logistic regression)



Rectified linear unit (ReLU) (hidden layers)



Neural Networks with Hidden Layers



Code Example: Binary Classification with Keras and PyTorch

For binary classification, we use binary cross entropy

Experiment 1, Keras: Jupyter Notebook:

https://github.com/pnugues/ilppp/blob/master/programs/ch04/keras/Salamambo_keras.ipynb

Experiment 1, PyTorch: Jupyter Notebook:

https://github.com/pnugues/ilppp/blob/master/programs/ch04/keras/Salamambo_torch.ipynb



Code Example: Multinomial Classification with Keras and PyTorch

For multinomial (multiclass) classification, we use (categorical) cross entropy

Experiment 1, Keras: Jupyter Notebook:

https://github.com/pnugues/ilppp/blob/master/programs/ch04/keras/Salamambo_multi_keras.ipynb

Experiment 1, PyTorch: Jupyter Notebook:

https://github.com/pnugues/ilppp/blob/master/programs/ch04/keras/Salamambo_multi_torch.ipynb



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.



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 $\frac{|AUD|}{|PUN|}$.

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

A task in natural language processing: Identify the parts of speech (POS) of words.

Example: *The can rusted*

- The human: *The*/art (DT) *can*/noun (NN) *rusted*/verb (VBD)
- The POS tagger: *The*/art (DT) *can*/modal (MD) *rusted*/verb (VBD)

| ↓Correct | Tagger → | | | | | | | | | |
|----------|----------|------|------|------|------|------|------|------|------|------|
| | DT | IN | JJ | NN | RB | RP | VB | VBD | VBG | VCN |
| 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 | – |
| VCN | – | – | 4.6 | – | – | – | – | 4.3 | – | 90.6 |

After Franz (1996, p. 124)