Machine Learning
Regression

Charlotte Pelletier
Univ. Bretagne Sud – IRISA Vannes
Based on C. Friguet's lecture.

28 September 2021

Content

Regression

Recall

- Modelling the relationship between the target variable y and the explanatory variables $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_d$
  - y is the **qualitative** (binary) variable to be explained
  - $\mathbf{X}_i$ are quantitative or qualitative explanatory variables

➜ The **linear model** is **not tailored**

Recall

- Modelling the relationship between the target variable y and the explanatory variables $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_d$
    - y is the **qualitative** (binary) variable to be explained
    - $\mathbf{X}_i$ are quantitative or qualitative explanatory variables

➜ The **linear model** is **not tailored** because the variable y is not quantitative!

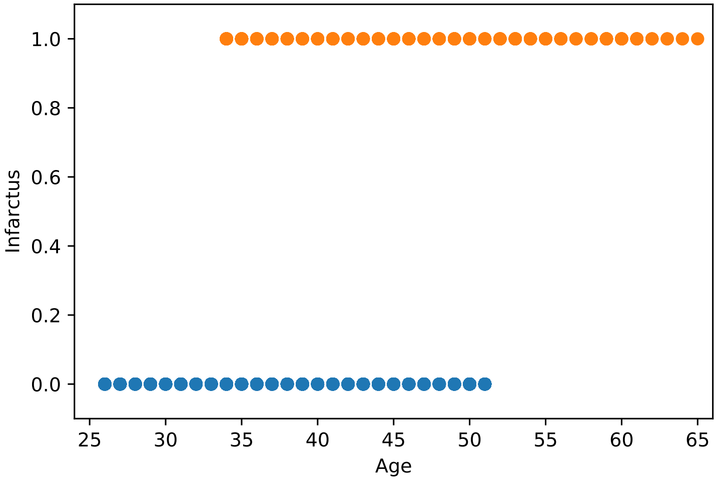
Example B

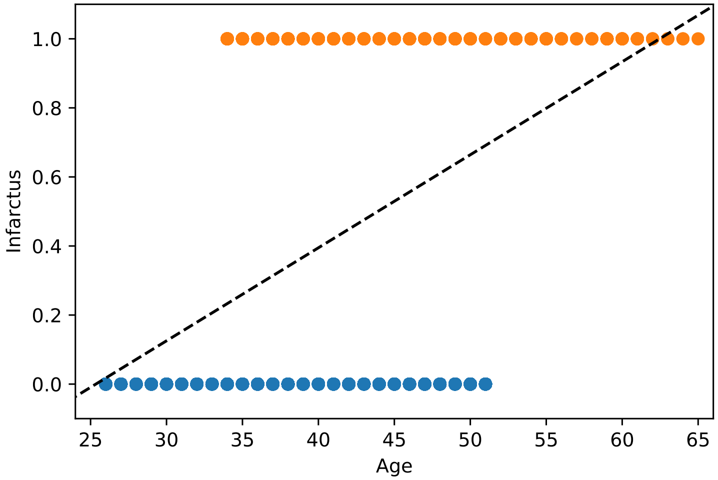
We want to evaluate the risk of myocardial infarction (heart attack) on women from a case-study. The collected factors are: taking contraceptive, age, weight, smoking, high blood pressure, a family history of cardiovascular disease, *etc*.

Data subset:

Infarction	Taking contraceptive	Age	Weight	...
no	no	47	48	...
no	no	35	53	...
no	yes	62	41	...
yes	yes	47	45	...
yes	yes	63	...	
yes	no	45	69	...
yes	no	90	84	...
⋮	⋮	⋮	⋮	⋮

Data from Institut de Santé Publique, d'Epidémiologie et de Développement (ISPED), Bordeaux
Source: `http://www.biostatisticien.eu/springeR/jeuxDonnees5.html`


1.0
0.8
0.6
0.4
0.2
0.0
Infarctus
25
30
35
40
45
50
55
60
65
Age



1.0
0.8
0.6
0.4
0.2
0.0
Infarctus
25
30
35
40
45
50
55
60
65
Age

Content

Some probabilities

Adjustment

- we do not modelled Y by a linear relationship
- we look at the probabilities: $\mathbf{P}(Y = 0|X = x)$ and $\mathbf{P}(Y = 1|X = x)$
- the knowledge of $\mathbf{P}(Y = 1|X = x)$ induces the knowledge of $\mathbf{P}(Y = 0|X = x)$ because

Some probabilities

Adjustment

- we do not modelled Y by a linear relationship
- we look at the probabilities: $\mathbf{P}(Y = 0|X = x)$ and $\mathbf{P}(Y = 1|X = x)$
- the knowledge of $\mathbf{P}(Y = 1|X = x)$ induces the knowledge of $\mathbf{P}(Y = 0|X = x)$ because $\mathbf{P}(Y = 0|X = x) = 1 - \mathbf{P}(Y = 1|X = x)$

Let us write out

$$\mathbf{P}(Y = 1|X = x) = \pi(x)$$

and

$$\mathbf{P}(Y = 0|X = x) = 1 - \pi(x),$$

where $\pi(x) \in [0, 1]$.

We can model Y by

$$Y|X = x \sim \mathcal{B}(\pi(x))$$

(Bernoulli's law)

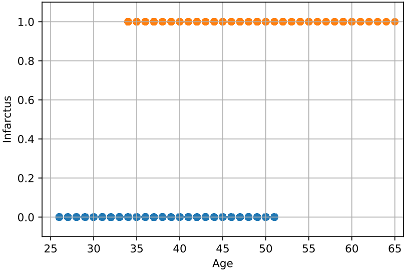
Properties (Bernoulli's law)

$$\mathbf{P}(Y = y|X = x) = \pi(x)^y \big(1 - \pi(x)\big)^{1-y}$$

$$\mathbb{E}(Y|X = x) = \pi(x) \qquad Var(Y|X = x) = \pi(x)\big(1 - \pi(x)\big)$$

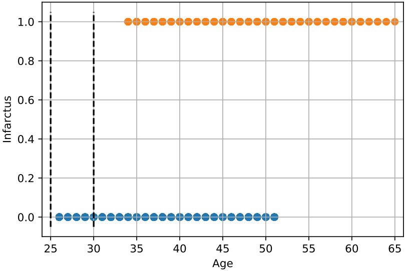
- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

categ.	n_c	#0	#1	$\hat{\pi}(x)$



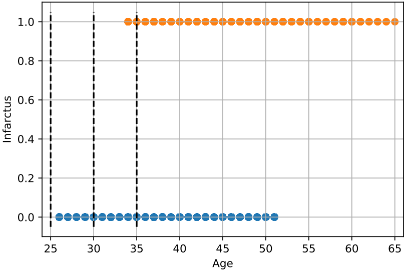
- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

categ.	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00



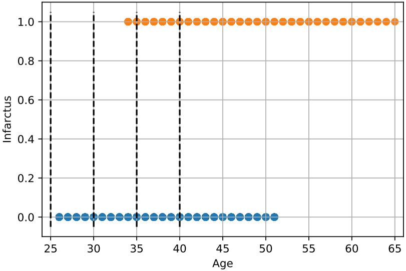
- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

categ.	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04



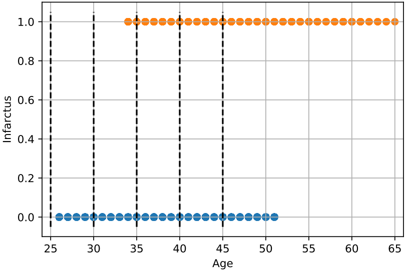
- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

categ.	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10



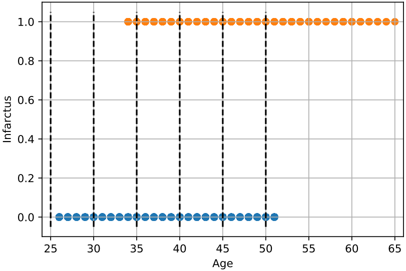
- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

categ.	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17



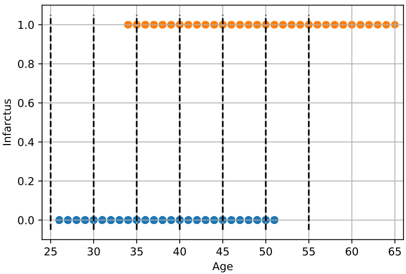
- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

categ.	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36



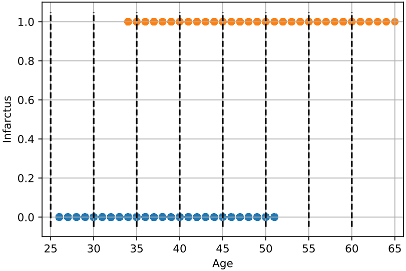
- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

categ.	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78



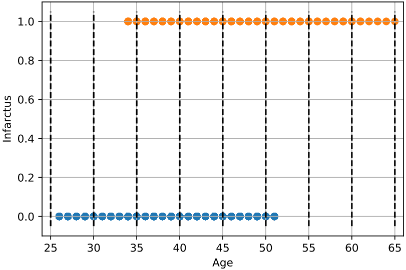
- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

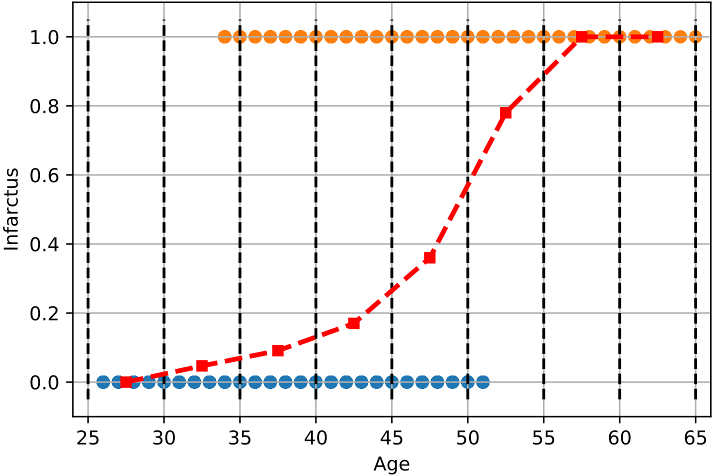
categ.	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78
(55,60]	33	0	33	1.00



- Estimation of $\pi(x)$: $\hat{\pi}(x)$ = proportion of $Y = 1$ for X divided into categories

categ.	n_c	#0	#1	$\hat{\pi}(x)$
[25,30]	454	454	0	0.00
(30,35]	958	913	45	0.04
(35,40]	780	709	71	0.10
(40,45]	608	505	103	0.17
(45,50]	554	355	199	0.36
(50,55]	134	30	104	0.78
(55,60]	33	0	33	1.00
(60,65]	17	0	17	1.00




1.0
0.8
0.6
0.4
0.2
0.0
Infarctus
25
30
35
40
45
50
55
60
65
Age

Logistic function

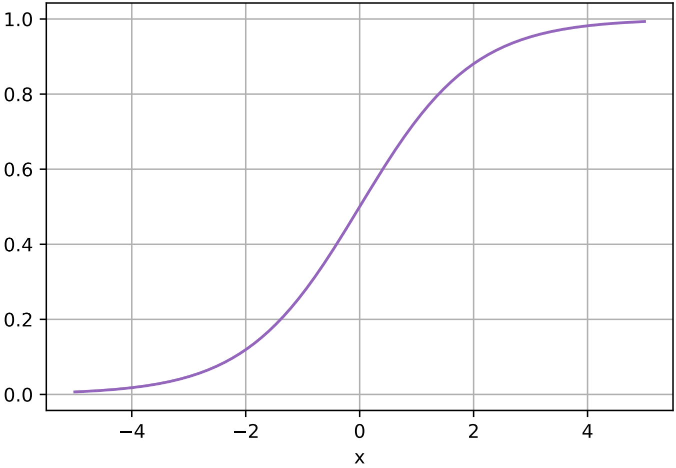
- $\pi(x)$ is not a linear function of x!
- We rather want to model $\pi(x)$ by a function with a $''\mathcal{S}''$ shape $\Rightarrow$ one possibility is the logistic function

Logistic function

$$\begin{aligned} s \quad : \quad & \mathbb{R} \to]0,1[\\ & x \to s(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x} \end{aligned}$$

It is a specific case of the sigmoid function.

Logistic function



Logistic function

Shape of the function curve (more general)

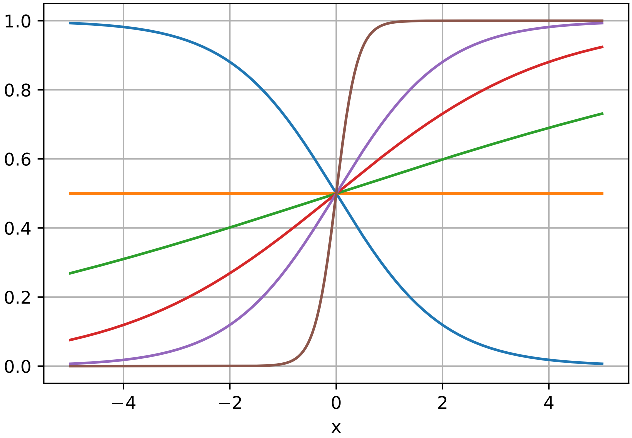
$$\begin{aligned} s \quad &: \quad \mathbb{R} \to]0,1[\\ & x \to s(x) = \frac{1}{1+e^{-x\beta}} = \frac{e^{x\beta}}{1+e^{x\beta}} \end{aligned}$$

for several β values

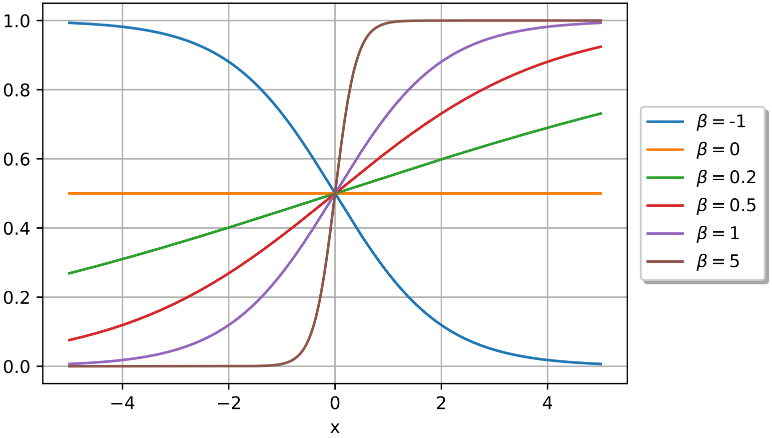
- $\beta = 0 \Rightarrow$ constant function
- "small" β values $\Rightarrow$ wide range of x values for which the function is around 0.5 $\Rightarrow$ difficult discrimination
- "big" β values $\Rightarrow$ small range of x values for which the function is around 0.5 $\Rightarrow$ easier discrimination

⚠ It depends on the scale of x!

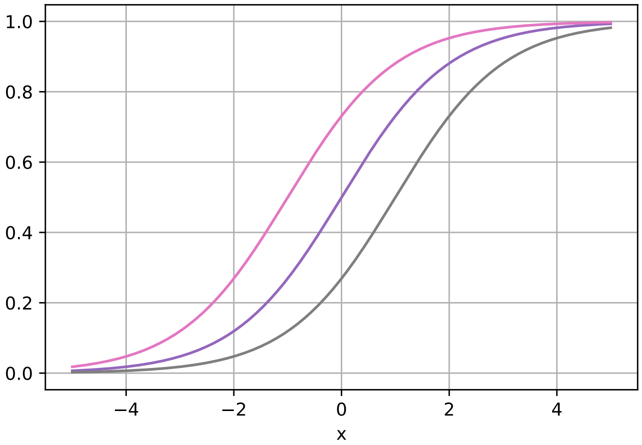
Logistic function



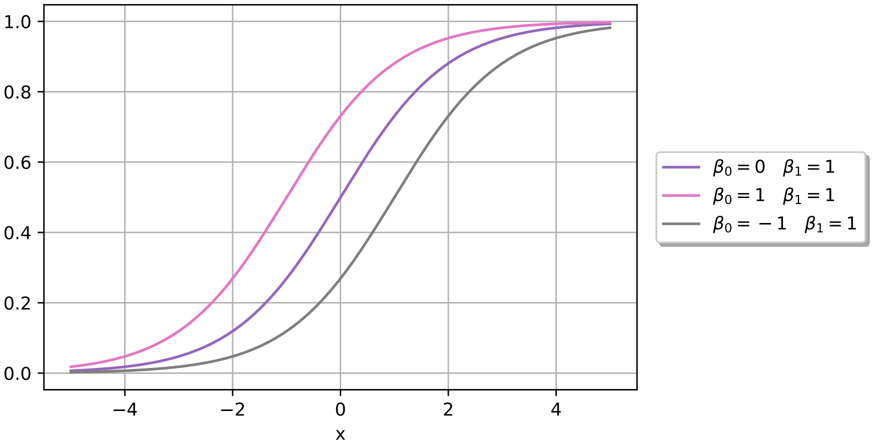
Logistic function



Logistic function



Logistic function



Logistic model

Logistic model

$$\pi(\mathbf{x}) = \frac{e^{\tilde{\mathbf{x}}\boldsymbol{\beta}}}{1+e^{\tilde{\mathbf{x}}\boldsymbol{\beta}}} = \frac{e^{\beta_0+\beta_1 x_1+\ldots+\beta_d x_d}}{1+e^{\beta_0+\beta_1 x_1+\ldots+\beta_d x_d}}$$

$$\Updownarrow$$

$$logit\Big(\pi(\mathbf{x})\Big) = log\left(\frac{\pi(\mathbf{x})}{1-\pi(\mathbf{x})}\right) = \tilde{\mathbf{x}}\boldsymbol{\beta} = \beta_0 + \beta_1 x_1 + \ldots + \beta_d x_d$$

- Transformation g that creates a linear link between $g\Big(\pi(\mathbf{x})\Big)$ and $\mathbf{x}$: **link function**

Logit function

$$\begin{aligned} logit \quad : \quad &]0,1[\to \mathbb{R} \\ & p \quad \to logit(p) = log\left(\frac{p}{1-p}\right) \end{aligned}$$

Logistic model

- Let us write out:

$$f_{\boldsymbol{\beta}}(\mathbf{x}_i) = \frac{e^{\tilde{\mathbf{x}}_i \boldsymbol{\beta}}}{1 + e^{\tilde{\mathbf{x}}_i \boldsymbol{\beta}}} = \mathbb{P}(Y = 1 | X = \tilde{\mathbf{x}}_i)$$

- We want $\boldsymbol{\beta}$ such that $f_{\boldsymbol{\beta}}(\mathbf{x}_i)$ is as closed as possible of y_i for all the training data $\{\mathbf{x}_i, y_i\}_{i=1}^{m}$ where $\mathbf{x}_i \in \mathbb{R}^d$

- Model likelihood:

$$L(\boldsymbol{\beta}, y) \quad = \quad \prod_{i=1}^{m} \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^{m} f_{\boldsymbol{\beta}}(\mathbf{x}_i)^{y_i} \times (1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i}$$

Logistic model

- Model likelihood:

$$L(\boldsymbol{\beta}, y) \quad = \quad \prod_{i=1}^{m} \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^{m} f_{\boldsymbol{\beta}}(\mathbf{x}_i)^{y_i} \times (1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i}$$

- Log-likelihood

$$\mathcal{L}(\boldsymbol{\beta}, y) \quad = \quad \sum_{i=1}^{m} \Big[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1 - y_i) log\Big(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big)\Big]$$

Logistic model

- Model likelihood:

$$L(\boldsymbol{\beta}, y) \quad = \quad \prod_{i=1}^{m} \mathbf{P}(Y = y_i | X = \mathbf{x}_i) = \prod_{i=1}^{m} f_{\boldsymbol{\beta}}(\mathbf{x}_i)^{y_i} \times (1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i))^{1-y_i}$$

- Log-likelihood

$$\mathcal{L}(\boldsymbol{\beta}, y) \quad = \quad \sum_{i=1}^{m} \left[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1 - y_i) log\Big(1 - f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) \right]$$

- Maximisation of (log-)likelihood
  $\Rightarrow$ cancel the derivatives of $L(\boldsymbol{\beta}, y)$
  - **no explicit analytical solution**
  - there is a need for optimisation iterative algorithms

Cost function

- **Objective**: find the best $\boldsymbol{\beta}$ to minimise the global (quadratic) cost:

$$\underset{\beta}{\operatorname{argmin}}\Big(\mathbf{J}(\beta)\Big)$$

where

$$\mathbf{J}(\boldsymbol{\beta}) = -\frac{1}{m}\sum_{i=1}^{m}\Big[y_i log\Big(f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big) + (1-y_i) log\Big(1-f_{\boldsymbol{\beta}}(\mathbf{x}_i)\Big)\Big]$$

Content

Principle

- Objective: find the minimum of the cost-function
- Principle: iterative algorithm
  1. initialisation: $\beta^{(0)}$
  2. at each step k, change the $\boldsymbol{\beta}^{(k-1)}$ value to decrease $\mathbf{J}(\boldsymbol{\beta}^{(k)})$
  3. stop the process when a minimum is reached

Iteration k of the gradient descent algorithm

For the β_j parameter

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} \mathbf{J}(\boldsymbol{\beta}^{(k-1)})$$

where:

- $\frac{\partial}{\partial \beta_j}$:
- α:

Principle

- Objective: find the minimum of the cost-function
- Principle: iterative algorithm
  1. initialisation: $\beta^{(0)}$
  2. at each step k, change the $\boldsymbol{\beta}^{(k-1)}$ value to decrease $\mathbf{J}(\boldsymbol{\beta}^{(k)})$
  3. stop the process when a minimum is reached

Iteration k of the gradient descent algorithm

For the β_j parameter

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{\partial}{\partial \beta_j} \mathbf{J}(\boldsymbol{\beta}^{(k-1)})$$

where:

- $\frac{\partial}{\partial \beta_j}$: partial derivatives
- α: learning rate (hyperparameter to be determined by the user)

For the logistic regression

- Iteration k of the gradient descent algorithm?

Iteration k of the gradient descent algorithm - logistic regression

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^{m} \Big(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i \Big) x_{ij}$$

Note: $\forall i, x_{i0} = 1$

For the logistic regression

- Iteration k of the gradient descent algorithm?
- Same formula than for the linear regression...

Iteration k of the gradient descent algorithm - logistic regression

$$\beta_j^{(k)} \quad := \quad \beta_j^{(k-1)} - \alpha \frac{1}{m} \sum_{i=1}^{m} \Big(f_{\boldsymbol{\beta}^{(k-1)}}(\mathbf{x}_i) - y_i \Big) x_{ij}$$

Note: $\forall i, x_{i0} = 1$

Optimisation

There are more advanced versions of the gradient descent algorithm:

- conjugated gradient, BFGS[1], (limited-memory) L-BFGS
  - no need to fix α (it is done automatically by the algorithm)
  - faster (it requires less iterations) than the classical gradient descent algorithm
  - but more complex to implement (these variants are avaible in most of the software/programming languages)

[1] Broyden-Fletcher-Goldfarb-Shanno