

Logistic regression

Gilles Gasso

INSA Rouen - ITI Department
Laboratory LITIS

November 6, 2023

Outline

- 1 Introduction
- 2 Logistic regression model
- 3 Parameters Estimation
 - Criterion
 - Estimation
 - Algorithm: a sketch
 - Prediction with the model
- 4 Extension to multi-class logistic regression
- 5 Conclusion

Discrimination and prior probability

Classify athletes using their biological measures: $\mathcal{D} = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \{0, 1\}\}_{i=1}^N$

Labels: let $y = 0$ for male and $y = 1$ for female athletes

rcc	wcc	hc	hg	ferr	bmi	ssf	pcBfat	lbm	ht	wt	sex
4.82	7.6	43.2	14.4	58	22.37	50	11.64	53.11	163.9	60.1	f
4.32	6.8	40.6	13.7	46	17.54	54.6	12.16	46.12	173	52.5	f
5.16	7.2	44.3	14.5	88	18.29	61.9	12.92	48.76	175	56	f
4.53	5	40.7	14	41	17.79	56.8	12.55	38.3	156.9	43.8	f
4.42	6.4	42.8	14.5	63	20.31	58.9	13.46	39.03	149	45.1	f
4.93	7.3	46.2	15.1	41	21.12	34	6.59	67	184.4	71.8	m
5.21	7.5	47.5	16.5	20	21.89	46.7	9.5	70	187.3	76.8	m
5.09	8.9	46.3	15.4	44	29.97	71.1	13.97	88	185.1	102.7	m
4.94	6.3	45.7	15.5	50	23.11	34.3	6.43	74	184.9	79	m
4.86	3.9	44.9	15.4	73	22.83	34.5	6.56	70	181	74.8	m
4.51	4.4	41.6	12.7	44	19.44	65.1	15.07	53.42	179.9	62.9	f
4.62	7.3	43.8	14.7	26	21.2	76.8	18.08	61.85	188.7	75.5	f

Inputs \mathbf{X}

Labels \mathbf{y}

What is the prior probability $\mathbb{P}(y = 1)$ that an athlete is a female?

Posterior probability , odds and score

- What is the probability that an athlete with known input \mathbf{x} is $y = 1$?

- Bayes' Decision : $D(\mathbf{x}) = \begin{cases} 1 & \text{if } \frac{\mathbb{P}(y=1/\mathbf{x})}{\mathbb{P}(y=0/\mathbf{x})} > 1 \\ 0 & \text{otherwise} \end{cases}$

Require the conditional distributions $p(\mathbf{x}/y = 1)$ and $p(\mathbf{x}/y = 0)$

- Bayes' Score :

$$\text{score}(\mathbf{x}) = \log \left(\frac{\mathbb{P}(y = 1/\mathbf{x})}{1 - \mathbb{P}(y = 1/\mathbf{x})} \right)$$

Logistic regression: motivation

- The decision rule only requires knowledge of the score

$$\text{score}(\mathbf{x}) = \log \left(\frac{\mathbb{P}(y = 1/\mathbf{x})}{1 - \mathbb{P}(y = 1/\mathbf{x})} \right)$$

- The decision function is $D(\mathbf{x}) = \text{sign}(\text{score}(\mathbf{x}))$

Goal of logistic regression

- Learn directly a scoring function $f(\mathbf{x})$

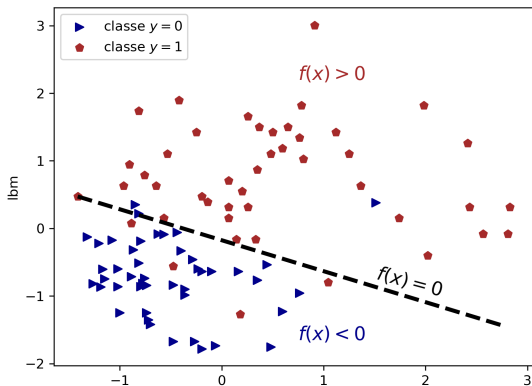
$$\text{score}(\mathbf{x}) = \log \left(\frac{\mathbb{P}(y = 1/\mathbf{x})}{1 - \mathbb{P}(y = 1/\mathbf{x})} \right) = f(\mathbf{x})$$

→ Avoid to learn the conditional distributions $p(\mathbf{x}/y)$ and the prior $\mathbb{P}(y)$ to get the posterior probabilities $\mathbb{P}(y/\mathbf{x})$

Scoring function

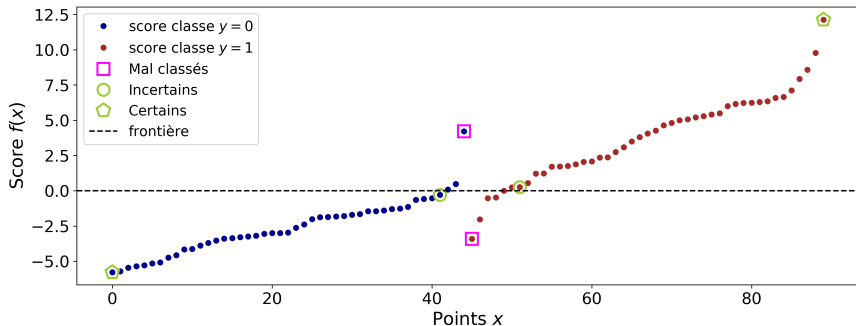
- Model : $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$
- Decision rule : assign \mathbf{x} to $\hat{y} = \begin{cases} 1 & \text{if } f(\mathbf{x}) \geq 0 \\ 0 & \text{if } f(\mathbf{x}) < 0 \end{cases}$

Athletes' classification problem using two variables ($ferr$, lbm)



Confidence in the decision making

Sort the score



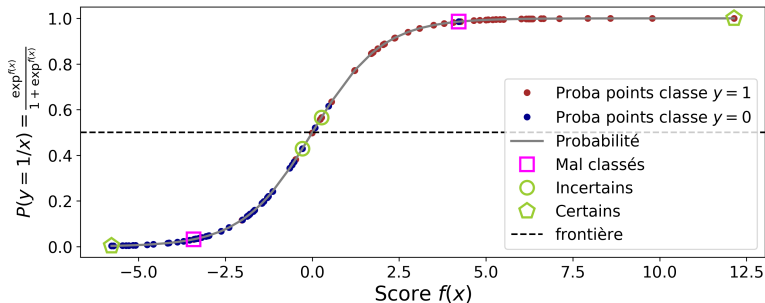
- Confident : $f(\mathbf{x}) \rightarrow \infty$ and $y = 1$ or $f(\mathbf{x}) \rightarrow -\infty$ and $y = 0$
- Uncertain : $f(\mathbf{x}) \rightarrow 0$

Quantify the confidence: from the score to posterior probability

- Use an increasing monotone function $\mathbb{R} \rightarrow [0, 1]$: sigmoid

$$\mathbb{P}(y = 1|\mathbf{x}) = \frac{\exp^{f(\mathbf{x})}}{1 + \exp^{f(\mathbf{x})}} \rightarrow \mathbb{P}(y = 0|\mathbf{x}) = 1 - \mathbb{P}(y = 1|\mathbf{x}) = \frac{1}{1 + \exp^{f(\mathbf{x})}}$$

- the decision function reads $\hat{y} = \begin{cases} 1 & \text{if } \mathbb{P}(y = 1|\mathbf{x}) > 0.5 \\ 0 & \text{if } \mathbb{P}(y = 1|\mathbf{x}) < 0.5 \end{cases}$



Estimate the scoring function f

- We seek f such that for any given training sample $\mathbf{x}_i \in \mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$

$$\mathbb{P}(y_i = 1 | \mathbf{x}_i) = \frac{\exp^{f(\mathbf{x}_i)}}{1 + \exp^{f(\mathbf{x}_i)}} \rightarrow \begin{cases} 1 & \text{if } y_i = 1 \\ 0 & \text{if } y_i = 0 \end{cases}$$

- Maximize the conditional log-likelihood

$$\begin{aligned} \mathcal{L}(\{y_i\}_{i=1}^N / \{\mathbf{x}_i\}_{i=1}^N; f) &= \log \prod_{i=1}^N [\mathbb{P}(y_i = 1 | \mathbf{x}_i)^{y_i} (1 - \mathbb{P}(y_i = 1 | \mathbf{x}_i))^{1-y_i}] \\ &= \sum_{i=1}^N y_i \log(\mathbb{P}(y_i = 1 | \mathbf{x}_i)) + (1 - y_i) \log(1 - \mathbb{P}(y_i = 1 | \mathbf{x}_i)) \end{aligned}$$

Relevant optimization problem

$$\max_f \mathcal{L}(\{y_i\}_{i=1}^N / \{\mathbf{x}_i\}_{i=1}^N; f) \Leftrightarrow \min_f J(f)$$

with $J(f) = -\sum_{i=1}^N [y_i \log(\mathbb{P}(y_i = 1 | \mathbf{x}_i)) + (1 - y_i) \log(1 - \mathbb{P}(y_i = 1 | \mathbf{x}_i))]$

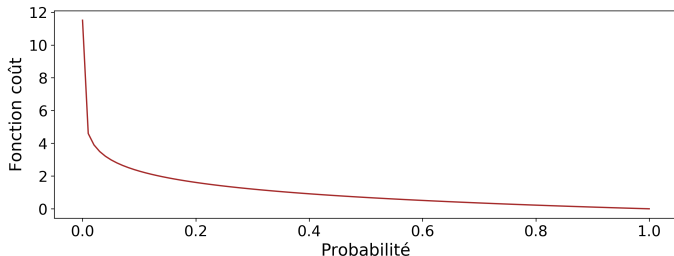
Fitting objective function

- Re-writing the criterion $J(f)$

$$J(f) = \sum_{i=1}^N \ell(y_i, p_i)$$

with $\ell(y_i, p_i) = -y_i \log p_i - (1 - y_i) \log(1 - p_i)$ and $p_i = \mathbb{P}(y_i = 1 | \mathbf{x}_i)$

- $\ell(y, p)$: loss function known as **binary cross entropy**



A brief summary

- Scoring function: $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = \boldsymbol{\varphi}^\top \boldsymbol{\theta}$

with $\boldsymbol{\varphi} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \in \mathbb{R}^{d+1}$ and $\boldsymbol{\theta} = \begin{pmatrix} \mathbf{w} \\ b \end{pmatrix} \in \mathbb{R}^{d+1}$ for $\mathbf{x} \in \mathbb{R}^d$

- Posterior probability: $\mathbb{P}(y = 1 | \mathbf{x}) = p = \frac{\exp^{f(\mathbf{x})}}{1 + \exp^{f(\mathbf{x})}} = \frac{\exp^{\boldsymbol{\varphi}^\top \boldsymbol{\theta}}}{1 + \exp^{\boldsymbol{\varphi}^\top \boldsymbol{\theta}}}$
- We deduce the optimization problem

$$\min_f J(f) = \sum_{i=1}^N -y_i \log p_i - (1 - y_i) \log(1 - p_i)$$

$$\Leftrightarrow \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \sum_{i=1}^N [-y_i \boldsymbol{\varphi}_i^\top \boldsymbol{\theta} + \log(1 + \exp^{\boldsymbol{\varphi}_i^\top \boldsymbol{\theta}})]$$

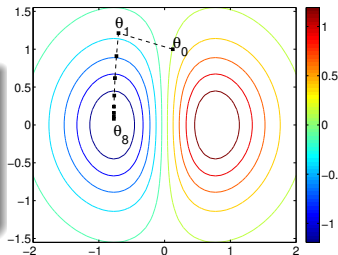
Compute the solution ...

... using a descent algorithm

Reminder: descent methods

Principle for solving $\min_{\theta} J(\theta)$

- Start from θ_0
- Build the sequence $\{\theta_k\}$ with
$$\theta_{k+1} = \theta_k + \alpha_k \mathbf{h}_k$$
- dots converging toward a stationary point $\hat{\theta}$



- \mathbf{h}_k : **descent direction** such that $J(\theta_k) > J(\theta_{k+1})$, α_k : **step size**

Examples

- Gradient descent method: $\mathbf{h} = -\nabla J(\theta)$
- Newton method: $\mathbf{h} = -\mathbf{H}^{-1} \nabla J(\theta)$ with \mathbf{H} the hessian matrix

Logistic regression: estimation of the parameters θ

Newton Algorithm applied to logistic regression

$$J(\theta) = \sum_{i=1}^N [-y_i \varphi_i^\top \theta + \log(1 + \exp^{\varphi_i^\top \theta})]$$

- Gradient $g = \nabla_{\theta} J(\theta)$

$$\begin{aligned} \nabla J(\theta) &= -\sum_{i=1}^N y_i \varphi_i + \sum_{i=1}^N \varphi_i \frac{\exp^{\varphi_i^\top \theta}}{1 + \exp^{\varphi_i^\top \theta}} \\ &= -\sum_{i=1}^N (y_i - p_i) \varphi_i \quad \text{with} \quad p_i = \frac{\exp^{\varphi_i^\top \theta}}{1 + \exp^{\varphi_i^\top \theta}} \end{aligned}$$

- Hessian matrix $H = \frac{\partial^2 J(\theta)}{\partial \theta \partial \theta^\top}$

$$H = \sum_{i=1}^N p_i (1 - p_i) \varphi_i \varphi_i^\top$$

Gradient and Hessian: matrix form

- Let

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix} \in \mathbb{R}^N, \quad \Phi = \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_N^\top \end{bmatrix} \in \mathbb{R}^{N \times (d+1)}$$

- Let $W \in \mathbb{R}^{N \times N}$ be the diagonal matrix so that

$$W_{ii} = p_i(1 - p_i)$$

- We can easily establish that

$$\begin{array}{ll} \text{gradient} & \mathbf{g} = -\Phi^\top (\mathbf{y} - \mathbf{p}) \\ \text{Hessian} & \mathbf{H} = \Phi^\top W \Phi \end{array}$$

Logistic regression: the iterates

- Newton's method compute the following iterates starting from θ_0

$$\theta_{k+1} = \theta_k - \mathbf{H}_k^{-1} \mathbf{g}_k$$

- The gradient and hessian at θ_k are given by

$$\begin{aligned} \mathbf{g}_k &= -\Phi^\top (y - p_k) \\ \mathbf{H}_k &= \Phi^\top \mathbf{W}_k \Phi \end{aligned}$$

where p_k and \mathbf{W}_k are computed based on $p_k = \frac{\exp \varphi^\top \theta_k}{1 + \exp \varphi^\top \theta_k}$

The Newton iterations

$$\longrightarrow \theta_{k+1} = \theta_k + (\Phi^\top \mathbf{W}_k \Phi)^{-1} \Phi^\top (y - p_k)$$

Algorithm

Input: data-set matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ and labels' vector $\mathbf{y} \in \mathbb{R}^N$

Output : parameters estimation vector $\boldsymbol{\theta}$

- 1 Form the matrix $\Phi = [\mathbf{I} \ \mathbf{X}]$
- 2 Initialization: set $k = 0$ and $\boldsymbol{\theta}_k = 0$.
- 3 Repeat

Form the vector \mathbf{p}_k st $p_k(i) = \frac{\exp \varphi_i^\top \boldsymbol{\theta}_k}{1 + \exp \varphi_i^\top \boldsymbol{\theta}_k}$, $i = 1, \dots, N$

Form the matrix $\mathbf{W}_k = \text{diag}(\tilde{\mathbf{p}}_k)$ where $\tilde{p}_k(i) = p_k(i)(1 - p_k(i))$

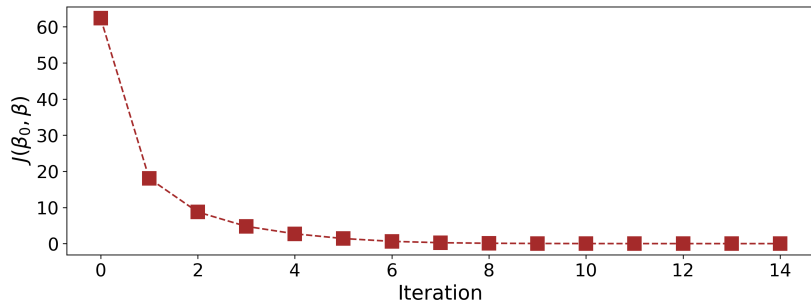
Calculate the new estimate

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \left(\Phi^\top \mathbf{W}_k \Phi \right)^{-1} \Phi^\top (\mathbf{y} - \mathbf{p}_k)$$

$k = k + 1$

- 4 Until convergence

Illustration



Remark

- In practice we solve the regularized optimisation problem

$$\min_{\theta} C J(\theta) + \Omega(\theta)$$

- $C > 0$: regularization parameter to be set by the user!
- Common regularisation : $\Omega(\theta) = \|\theta\|_2^2 = \sum_{j=1}^{d+1} \theta_j^2$
- To perform variable selection, choose: $\Omega(\theta) = \|\theta\|_1 = \sum_{j=1}^{d+1} |\theta_j|$

Predicting using the model

Classifying a new sample \mathbf{x}_j

- Given the parameters estimation $\hat{\theta}$
- Compute the posterior probabilities

$$\mathbb{P}(y = 1|\mathbf{x}_j) = \frac{\exp^{\varphi_j^\top \hat{\theta}}}{1 + \exp^{\varphi_j^\top \hat{\theta}}} \quad \text{and} \quad \mathbb{P}(y = 0|\mathbf{x}_j) = \frac{1}{1 + \exp^{\varphi_j^\top \hat{\theta}}}$$

$$\text{with } \varphi_j = \begin{pmatrix} \mathbf{x}_j \\ 1 \end{pmatrix}$$

- Predict label $\hat{y}_j = 1$ if $\mathbb{P}(y = 1|\mathbf{x}_j) \geq 1/2$ or $\hat{y}_j = 0$ otherwise

Extension to multi-class classification

We have K classes i.e. $y \in \{0, \dots, K-1\}$, $K > 2$

- We should determine $K-1$ scoring functions f_k
- The posterior probabilities are defined as

$$\mathbb{P}(y = k|\mathbf{x}) = \frac{\exp\varphi^\top \theta_k}{1 + \sum_{k=1}^{K-1} \exp\varphi^\top \theta_k} \quad \forall k = 1, \dots, K-1$$

$$\mathbb{P}(y = 0|\mathbf{x}) = \frac{1}{1 + \sum_{k=1}^{K-1} \exp\varphi^\top \theta_k}$$

- Decision rule:

predict the label with the maximum posterior probability i.e.

$$\hat{y} = \operatorname{argmax}_{k \in \{0, \dots, K-1\}} \mathbb{P}(y = k|\mathbf{x})$$

Multi-class logistic regression: estimation of the parameters

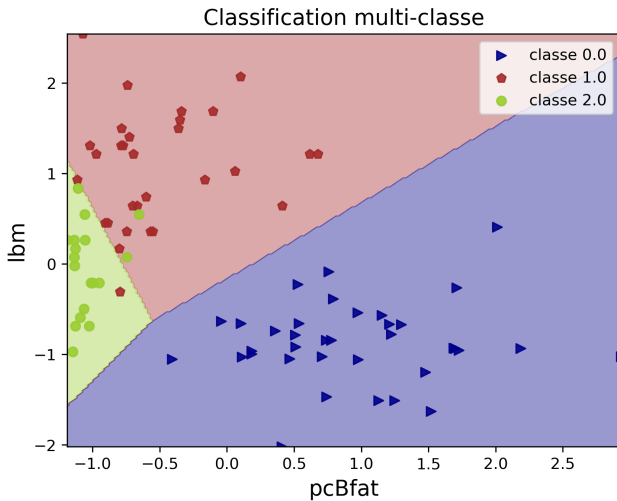
- Data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with $y_i \in \{0, \dots, K-1\}$
- Let define the **one-hot encoding vector** $\mathbf{z}_i \in \mathbb{R}^K$ with $z_{ik} = 1$ if $y_i = k$ and $z_{ik} = 0$ otherwise
- Example: for $K = 3$ and $y_i = 1$, we get $\mathbf{z}_i = (0 \ 1 \ 0)^\top$
- Conditional log-likelihood (multinomial distribution)

$$\mathcal{L} = \sum_{i=1}^N \sum_{k=1}^K z_{ik} \log \mathbb{P}(y = k | \mathbf{x}_i) \quad \text{with} \quad \mathbb{P}(y = k | \mathbf{x}) = \frac{\exp \varphi^\top \theta_k}{1 + \sum_{k=1}^{K-1} \exp \varphi^\top \theta_k}$$

Estimation of the parameters

Maximize the log-likelihood w.r.t. $K-1$ parameter vectors $\theta_1, \dots, \theta_{K-1}$

Illustration



Recapitulation

- Bayesian Decision Theory
 - Cost 0 - 1 : decision based on the comparison of the posterior probabilities
 - but it requires knowledge of conditional distributions and prior probabilities
- Logistic regression
 - directly models the probability ratio
 - knowledge of distributions is no longer needed
- Model Parameters Estimation
 - Maximisation of the log-Likelihood ...
 - ... by Newton's method

Conclusion

- simple (linear) model that yields to good prediction
- widely used model in several application (fraud detection, scoring)
- decision probabilities can be retrieved
- non-linear versions can be easily implemented

To find out more :

<http://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf>

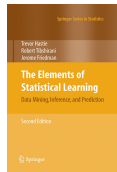
<http://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-modlin-reglog.pdf>

<https://stat.duke.edu/courses/Spring13/sta102.001/Lec/Lec20.pdf>

<http://www.cs.berkeley.edu/~russell/classes/cs194/f11/lectures/CS194%20Fall%202011%20Lecture%2006.pdf>

Python : [http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

[LogisticRegression.html](https://cran.r-project.org/web/packages/HSAUR/) R : <https://cran.r-project.org/web/packages/HSAUR/>



and book pages 120 and 121...