Logistic regression for classification problems

Gilles Gasso

INSA Rouen - ITI Department
Laboratory LITIS

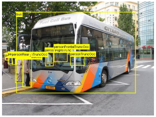
October 13, 2024

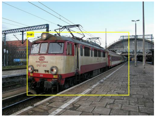
Outline

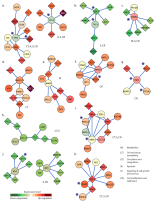
Classification problems

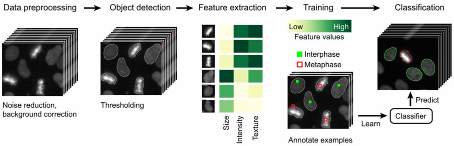
Applications

- Protein classification, Medical imaging
- Intrusion detection, fraud detection
- Object detection
- ...





Classification: taxonomy and formulation

- Data: $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$
- $\boldsymbol{x}$: sample belonging to the space $\mathcal{X}$ ($\mathcal{X} = \mathbb{R}^d$)
- $y \in \mathcal{Y}$: associated label with $\mathcal{Y}$: discrete finite set

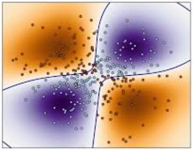
Taxonomy

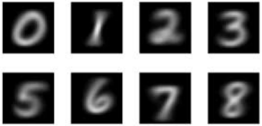
- Binary : $\mathcal{Y} = \{-1, 1\}$ ou $\mathcal{Y} = \{0, 1\}$

  Anomaly detection, Fraud detection ...
- Multi-class: $\mathcal{Y} = \{1, 2, \cdots, K\}$

  Objects or speakers recognition ...
- Multi-label: $\mathcal{Y} = 2^{\{1,2,\cdots,K\}}$

  Recognition of the topic of documents ...



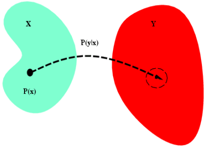


Classification: taxonomy and formulation

- Data: $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$
- $\boldsymbol{x}$: sample belonging to the space $\mathcal{X}$ ($\mathcal{X} = \mathbb{R}^d$)
- $y \in \mathcal{Y}$: associated label with $\mathcal{Y}$: discrete finite set

Principle

- Learn a mathematical function $f : \mathcal{X} \to \mathcal{Y}$ able to predict the label of $\boldsymbol{x}$
- Example: $\mathcal{Y} = \{-1, 1\}$ and the prediction function is $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$



Different approaches and algorithms

- Logistic regression, k-nearest neighbors, SVM, random forest, XGBoost, Deep Networks, ...

This lecture

- Logistic regression

Pre-requisites

Basics of probability and optimization

Discrimination and prior probability

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

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 $\boldsymbol{X}$ Labels $\boldsymbol{y}$

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

Posterior probability and decision

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

Statistical modeling of the data

- Conditional distributions: $p(x/y = 0)$ and $p(x/y = 1)$
- Marginal : $p_X(x) = p(x/y = 0)\mathbb{P}(y = 0) + p(x/y = 1)\mathbb{P}(y = 1)$

Decision

- Posterior probabilities
  $\mathbb{P}(y = 1/x) = \frac{p(x/y=1)\mathbb{P}(y=1)}{p_X(x)}, \quad \mathbb{P}(y = 0/x) = \frac{p(x/y=O)\mathbb{P}(y=0)}{p_X(x)}$
- Decision : $D(\boldsymbol{x}) = \begin{cases} 1 & \text{if} \quad \frac{\mathbb{P}(y=1/\boldsymbol{x})}{\mathbb{P}(y=0/\boldsymbol{x})} > 1 \\ 0 & \text{otherwise} \end{cases}$

Issue

Finding the conditional distributions $p(\boldsymbol{x}/y = 1)$ and $p(\boldsymbol{x}/y = 0)$ is hard

Posterior probability, odds and score

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

- Recall that the Decision is: $D(\boldsymbol{x}) = \begin{cases} 1 & \text{if} \quad \frac{\mathbb{P}(y=1/\boldsymbol{x})}{\mathbb{P}(y=0/\boldsymbol{x})} > 1 \\ 0 & \text{otherwise} \end{cases}$

  Requires the conditional distributions $p(\boldsymbol{x}/y = 1)$ and $p(\boldsymbol{x}/y = 0)$ (generally unknown)

- Odds: $\frac{\mathbb{P}(y=1/\boldsymbol{x})}{1-\mathbb{P}(y=1/\boldsymbol{x})}$

- Score:

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

Logistic regression: motivation

- The decision rule only requires knowledge of the score

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

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

Goal of logistic regression

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

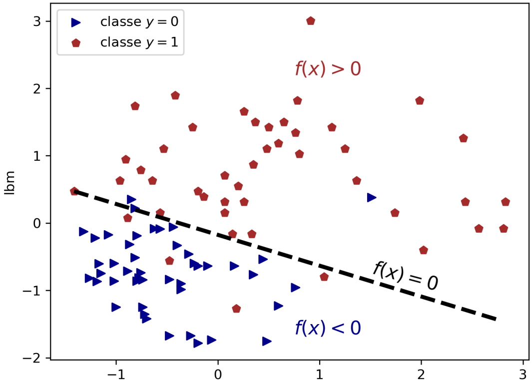
$$score(\boldsymbol{x}) = \log\left(\frac{\mathbb{P}(y=1/\boldsymbol{x})}{1-\mathbb{P}(y=1/\boldsymbol{x})}\right) = f(\boldsymbol{x})$$

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

Scoring function

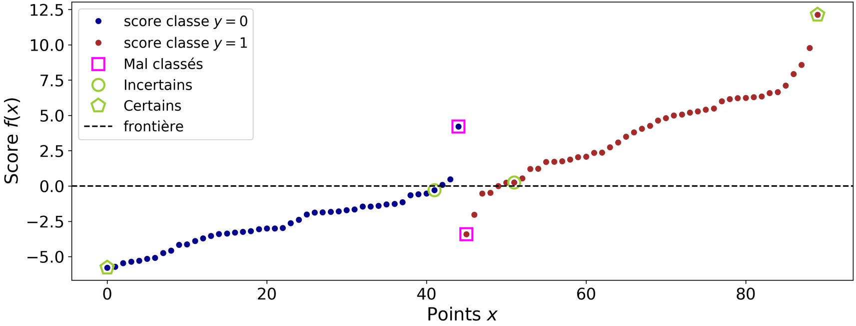
- Model: $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$
- Decision rule: assign $\boldsymbol{x}$ to $\hat{y} = \begin{cases} 1 & \text{if} \quad f(\boldsymbol{x}) \geq 0 \\ 0 & \text{if} \quad f(\boldsymbol{x}) < 0 \end{cases}$

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



Confidence in the decision making

Sort the score



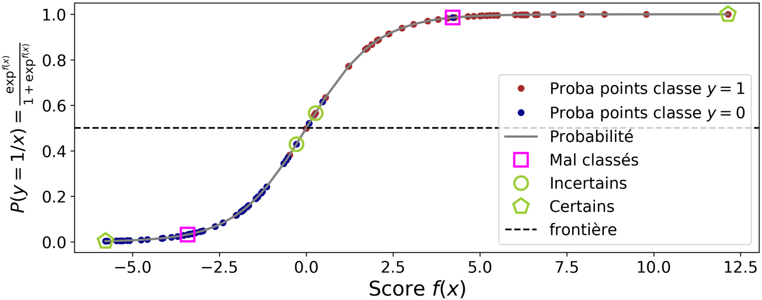
- Confident : $f(\boldsymbol{x}) \to \infty$ and $y = 1$ or $f(\boldsymbol{x}) \to -\infty$ and $y = 0$
- Uncertain : $f(\boldsymbol{x}) \to 0$

Quantify the confidence: from the score to posterior probability

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

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

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



Estimate the scoring function f

- We seek f such that for any given training sample $\boldsymbol{x}_i \in \mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$
  $\mathbb{P}(y_i = 1|\boldsymbol{x}_i) = \frac{\exp^{f(\boldsymbol{x}_i)}}{1+\exp^{f(\boldsymbol{x}_i)}} \rightarrow \begin{cases} 1 & \text{if} \quad y_i = 1 \\ 0 & \text{if} \quad y_i = 0 \end{cases}$

- Maximize the conditional log-likelihood

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

Relevant optimization problem

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

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

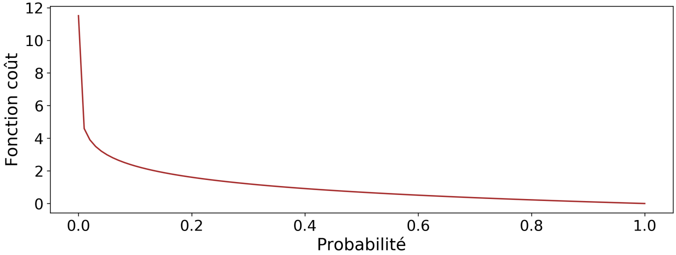
Fitting objective function

- 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 $p_i = \mathbb{P}(y_i = 1 | \boldsymbol{x}_i)$

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



A brief summary

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

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

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

- We deduce the optimization problem

$$\begin{aligned} \min_f J(f) &= \sum_{i=1}^{N} -y_i \log p_i - (1-y_i)\log(1-p_i) \\ \Leftrightarrow \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \left[-y_i \boldsymbol{\varphi}_i^\top \boldsymbol{\theta} + \log(1+\exp^{\boldsymbol{\varphi}_i^\top \boldsymbol{\theta}}) \right] \end{aligned}$$

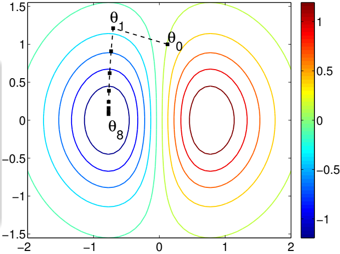
Compute the solution . . .

. . . using a descent algorithm

Reminder: descent methods

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

- Start from $\boldsymbol{\theta}_0$
- Build the sequence $\{\boldsymbol{\theta}_k\}$ with $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha_k \boldsymbol{h}_k$
- dots converging toward a stationary point $\hat{\boldsymbol{\theta}}$



- $\boldsymbol{h}_k$: descent direction such that $J(\boldsymbol{\theta}_k) < J(\boldsymbol{\theta}_{k+1})$, α_k: step size

Examples

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

Logistic regression: estimation of the parameters $\boldsymbol{\theta}$

Newton Algorithm applied to logistic regression

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

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

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

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

$$H = \sum_{i=1}^{N} p_i\, (1 - p_i)\, \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^\top$$

Gradient and Hessian: matrix form

- Let

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

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

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

- We can easily establish that

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

Logistic regression: the iterates

- Newton's method compute the following iterates starting from $\boldsymbol{\theta}_0$

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \boldsymbol{H}_k^{-1} \mathrm{g}_k$$

- The gradient and hessian at $\boldsymbol{\theta}_k$ are given by

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

where p_k and W_k are computed based on $p_k = \frac{\exp^{\boldsymbol{\varphi}^\top \boldsymbol{\theta}_t}}{1+\exp^{\boldsymbol{\varphi}^\top \theta_k}}$

The Newton iterations

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

Algorithm

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

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

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

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

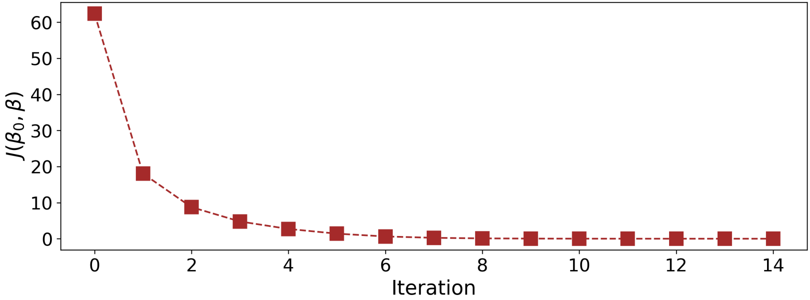
   Form the matrix $\mathrm{W}_k = \mathrm{diag}(\tilde{\mathrm{p}}_k)$ where $\tilde{\mathrm{p}}_k(i) = \mathrm{p_k}(i)(1 - \mathrm{p_k}(i))$

   Calculate the new estimate

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

   $k = k + 1$
4. Until convergence

Illustration



Remark

- In practice we solve the regularized optimisation problem

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

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

Predicting using the model

Classifying a new sample $\boldsymbol{x}_j$

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

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

with $\boldsymbol{\varphi}_j = \begin{pmatrix} \boldsymbol{x}_j \\ 1 \end{pmatrix}$

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

Extension to multi-class classification

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

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

$$\begin{aligned} \mathbb{P}(y = k|\boldsymbol{x}) &= \frac{\exp^{\boldsymbol{\varphi}^\top \boldsymbol{\theta}_k}}{1 + \sum_{k=1}^{K-1} \exp^{\boldsymbol{\varphi}^\top \boldsymbol{\theta}_k}} \quad \forall k = 1, \cdots, K-1 \\ \mathbb{P}(y = 0|\boldsymbol{x}) &= \frac{1}{1 + \sum_{k=1}^{K-1} \exp^{\boldsymbol{\varphi}^\top \boldsymbol{\theta}_k}} \end{aligned}$$

- Decision rule:

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

$$\hat{y} = \text{argmax}_{k \in \{0, \cdots, K-1\}} \mathbb{P}(y = k|\boldsymbol{x})$$

Multi-class logistic regression: estimation of the parameters

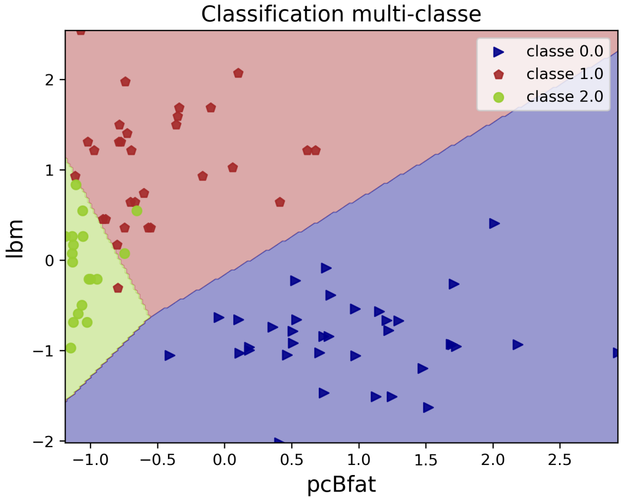
- Data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ with $y_i \in \{0, \cdots, K-1\}$
- Let define the **one-hot encoding vector** $\mathrm{z}^{(i)} \in \mathbb{R}^K$ with $z_k^{(i)} = 1$ if $y_i = k$ and $z_k^{(i)} = 0$ otherwise
- Example: for $K = 3$ and $y_i = 1$, we get $\mathrm{z}^{(i)} = \begin{pmatrix}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

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

Estimation of the parameters

Maximize the log-likelihood w.r.t. $K-1$ parameter vectors $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{K-1}$

Illustration



Summary

- Logistic regression
  - directly models the posterior probability ratio by a scoring function
  - The posterior probabilities can be retrieved from the scoring function
- Model Parameters Estimation
  - Maximisation of the log-Likelihood ...
  - ... by Newton's method
  - In practice a regularization scheme (often ℓ_1 or ℓ_2 norm of the parameters) is applied ...
  - ... and dedicated solving algorithms exist

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.LogisticRegression.html R : https://cran.r-project.org/web/packages/HSAUR/

and book pages 120 and 121...