Stochastic Machine Learning
01 - Introduction

Thorsten Schmidt

Abteilung für Mathematische Stochastik

www.stochastik.uni-freiburg.de
thorsten.schmidt@stochastik.uni-freiburg.de

WS 2020/21

Logistic regression

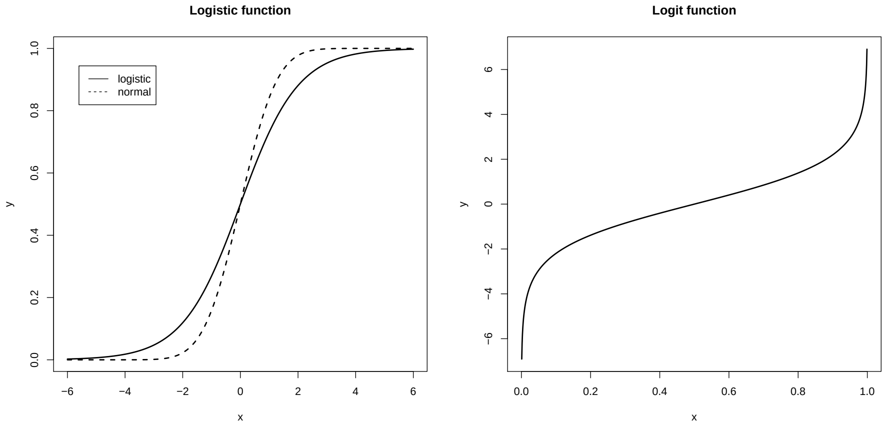
- One important regression approach for **classification** is logistic regression.
- We start by considering **simple** logistic regression, i.e. the classification into **two** classes. In this case, the response is always binary.
- One therefore needs to transform the whole real line to $[0, 1]$ and two approaches are common: first, via the logistic function

$$\sigma(x) = \frac{e^x}{1 + e^x}.$$

  The most common way is to transform y via

$$\sigma^{-1}(p) = \mathrm{logit}(p) = \log \frac{p}{1 - p},$$

  the so-called **logit** function.
- Second, by a cumulative distribution function (when this is Φ - standard normal - this approach is called **probit** model.


Logistic function
logistic
normal
y
x
0.0
0.2
0.4
0.6
0.8
1.0
−6
−4
−2
0
2
4
6
Logit function
y
x
−6
−4
−2
0
2
4
6
0.0
0.2
0.4
0.6
0.8
1.0

Definition (Logistic regression)

A logistic regression is the generalized linear model where

$$\operatorname{logit}(p_i) = \alpha + \boldsymbol{\beta}^\top \boldsymbol{x}_i, \qquad i = 1, \dots, n.$$

Note that this model is equivalent to

$$p_i = \frac{\exp(\alpha + \boldsymbol{\beta}^\top \boldsymbol{x}_i)}{1 + \exp(\alpha + \boldsymbol{\beta}^\top \boldsymbol{x}_i)}.$$

The observations $y_1, \dots, y_n$ are binary, hence take values in $\{0, 1\}$ and are assumed to be i.i.d. Bernoulli with $P(y_i = 1) = p_i = p_i(\boldsymbol{x}_i)$.
A nice source explaining the depth of logistic regression and their various applications is[13].

[13] Ronald Christensen (2006). **Log-linear models and logistic regression**. Springer Science & Business Media.

The most common estimation method used is **maximum-likelihood**. We take a small detour towards this exciting statistical concept going back to Sir Ronald Fisher.

Maximum-likelihood

$\mathcal{N}(\underbrace{\mu, \sigma^2}_{\theta})$ $\theta \in \Theta = \mathbb{R}, \mathbb{R}_{>0}$

- A **statistical model** is given by a family of probability measures $(P_\theta)_{\theta \in \Theta}$ on a common measurable space $(\Omega, \mathscr{F})$. It is typically called **parametric**, if Θ is of finite dimension.
- The **likelihood**-function for the observation E is given by
$$L(\theta) = P_\theta(E)$$
If $P_\theta(E) = 0$ for all $\theta \in \Theta$ one proceeds via the density: assume $P_\theta \ll P^*$ for all $\theta \in \Theta$ and denote the densities by $f_\theta := {}^{dP_\theta}/_{dP^*}$. Then, for the observation x,
$$L(\theta) = f_\theta(x).$$
- This looks complicated, but is in most cases quite simple: consider i.i.d. random variables $X_1, \ldots, X_n$ with common density f_θ. Then P^* is clearly the Lebesgue-measure. Due to the i.i.d.-property,
$$L(\theta) = \prod_{i=1}^{n} f_\theta(x_i).$$

Definition

Any maximizer $\hat{\theta}$ of the likelihood-function is called maximum-likelihood estimator for the model $(P_\theta)_{\theta \in \Theta}$.

In the above example, we need to maximize $\prod_{i=1}^n f_\theta(x_i)$, which is typically infeasible. One therefore considers the log-likelihood function

$$\ell(\theta) := \ln L(\theta)$$

which is often much easier to maximize. Typically one can apply first-order conditions or needs to solve numerically.

Example (ML for the normal distribution)

Consider $X_i \sim \mathcal{N}(\mu, 1)$. Then the density is

$$f_\theta(x) = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2}(x-\mu)^2 \Big).$$

$\Theta = \mu$

We obtain the log-likelihood function

$$l(\theta) = \text{const.} - \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2.$$

The first derivative is

$$\partial_\mu l(\theta) = \sum_{i=1}^{n} x_i - n\mu \overset{!}{=} 0$$

and we obtain the maximum-likelihood estimator (second derivative is < 0)

$$\hat{\mu} = \bar{x} = \frac{\sum_{i=1}^{n}}{n} x_i.$$

Exercise: compute the ML estimator for σ ! Read Czado & Schmidt (2011) on ML-estimation and further estimation procedures.

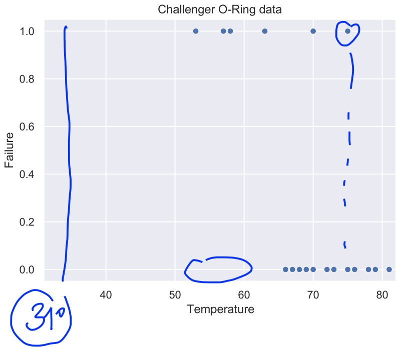
Maximum-Likelihood for the logistic regression

- For the logistic regression, where $y_1, \ldots, y_n$ are Bernoulli, we obtain the likelihood function
$$L(\boldsymbol{y}) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1-y_i}.$$
- Maximization has to be done numerically, eg. by gradient descent or by weighted least squares.
- Asymptotic distributions are available, such that we can test approximately several hypothesis, like for example $\beta_i = 0$ or $\alpha = 0$.

Back to logistic regression. We look at the by now infamous Challenger[14] O-ring data set (taken from Caslla & Berger (2002))

1	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**
53	**57**	**58**	**63**	**66**	**67**	**67**	**67**	**68**	**69**	**70**	**70**	**70**	**70**	**72**	**73**	**75**	**75**

The table reports failures with associated temperature.



[14]See https://en.wikipedia.org/wiki/Space_Shuttle_Challenger_disaster.

temp

fail

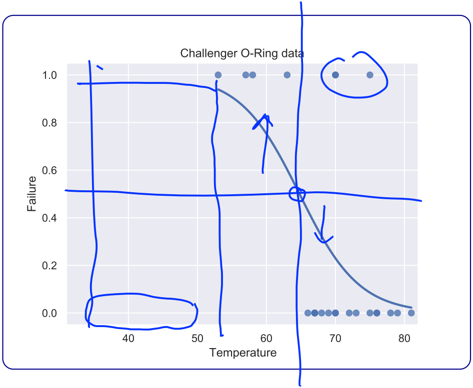
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression

x = np.array([53,57,58,63,66,67,67,67,68,69,70,70,70,70,72,73,
              75,75,76,76,78,79,81]).reshape(-1, 1)
y = np.array([1,  1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
              0, 1, 0, 0, 0, 0, 0]).reshape(-1, 1)

# logistic regression model
logreg = LogisticRegression().fit(x, y)

print(logreg.intercept_, logreg.coef_[0])
#[0.52055518] [-0.02100215]

import seaborn as sns
sns.set_theme(color_codes=True)
sns.regplot(x=x, y=y, logistic=True)
plt.show()
```



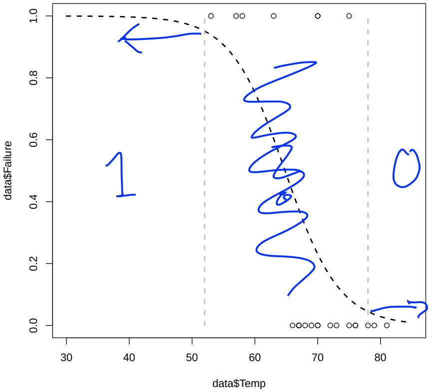
The estimated probability
for a failure at 31 degree is 0.9996088.

- Logistic regression naturally classifies the data into two fields: the ones with probability above 0.5, where we would optimally decide for outcome one and the ones with probability below 0.5, where we would decide for outcome 0.
- Hence, we obtain a **decision boundary**, given by the hyperplane

$$\alpha + \boldsymbol{\beta x} = 0.$$

- If the decision boundary separates the two groups, then the data is called **linearly separable.** Note that this can not be achieved in the Challenger dataset.
- Note that the logistic regression also provides probabilities of false decisions: at the boundary this is 50/50, but further out the probability of a false decision decrease. **Significant** decisions requires the probability of a false decision to be below a significance level, e.g. $\alpha = 0.05$ or $\alpha = 0.01$.

With significance level $\alpha = 0.05$ obtained decision boundaries.



Load the python example[15] from the homepage and revisit the above steps. Try your own examples.

- The likelihood-function has to be maximized numerically.
- A first-order iterative scheme is the **gradient-descent** algorithm. Look this algorithm up and recall its properties and functionality.

[15] Called 01_05_logistic_regression.py

Questions

- What is the difference between logistic regression and regression?
- What are the logit and probit functions ?
- What is maximum-likelihood?
- Compute the maximum-likelihood estimator for an exponential distribution.
- Look up the challenger catastrophe and watch Richard Feynman's famous speach.

Multi-classification