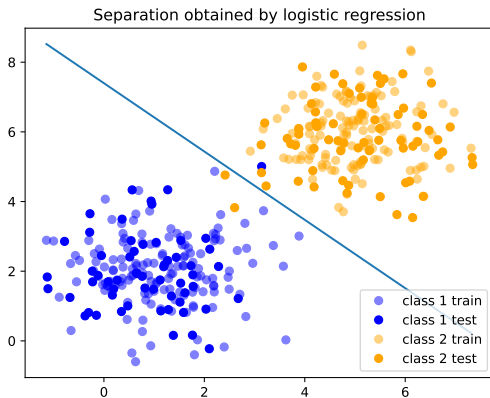


Machine learning I, supervised learning: logistic regression



General classification problem

- ▶ $\mathcal{X} = \mathbb{R}^d$
- ▶ $\mathcal{Y} = \{-1, 1\}$ or $\mathcal{Y} = \{0, 1\}$.
- ▶ $l(y, z) = 1_{y \neq z}$ ("0-1" loss)
https://en.wikipedia.org/wiki/Indicator_function
- ▶ $F = \mathcal{Y}^{\mathcal{X}}$

Problem

Optimizing on $F = \mathcal{Y}^{\mathcal{X}}$ (the set of all mappings from \mathcal{X} to \mathcal{Y}) is tricky because we cannot differentiate on this hypothesis space and it is not clear how to regularize.

Exercice 1 : Combinatorial problem

If we wanted to try all mappings in $\mathcal{Y}^{\mathcal{X}}$, if $|\mathcal{X}| = n$, how many mappings would there be?

Using scikit

Exercise 2 :

Use scikit to perform a logistic regression on the **digits** dataset.

template : `logistic_regression/logistic_regression_digits.py`

Real-valued function

Instead of an application in $\mathcal{Y}^{\mathcal{X}}$, we will learn $g : \mathcal{X} \rightarrow \mathbb{R}$ and define $f(x) = \text{sign}(g(x))$ with

$$\text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Risk

The risk (generalization error) of $f = \text{sign} \circ g$ writes :

$$\begin{aligned} R(g) &= P(\text{sign}(g(x)) \neq y) \\ &= E \left[1_{\text{sign}(g(x)) \neq y} \right] \\ &= E \left[1_{yg(x) < 0} \right] \end{aligned} \tag{1}$$

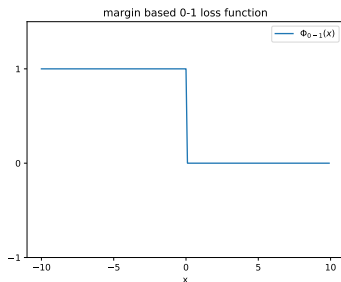
Several solutions

There might be many optimal maps g , i.e : such that $\text{sign}(g(x)) = f^*(x)$ for all x .

(Here, $f^*(x)$ is the Bayes predictor, which is the optimal predictor : it minimizes the generalization error (risque réel))

Margin based 0-1 loss function Φ_{0-1}

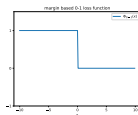
$$\begin{aligned} R(g) &= E \left[1_{\text{sign}(g(x)) \neq y} \right] \\ &= E \left[1_{yg(x) < 0} \right] \\ &= E \left[\Phi_{0-1}(yg(x)) \right] \end{aligned} \tag{2}$$



Empirical risk minimization

The corresponding empirical risk writes :

$$\frac{1}{n} \sum_{i=1}^n \Phi_{0-1}(y_i g(x_i)) \quad (3)$$



Issue with this objective function ?

- ▶ non-convex
- ▶ not continuous

Convex surrogate

Key idea : replace Φ_{0-1} by another function Φ that is easier to optimize (convexity) but still represents the correctness of the classification.

Natural question : but does minimizing the Φ -risk lead to a good "0-1" loss prediction ? Answering this question requires an advanced study.

Most common convex surrogates

Définition

Logistic loss

$$\Phi(u) = \log(1 + e^{-u}) \quad (4)$$

With linear predictors ($g(x_i) = \langle \theta, x_i \rangle$), this loss will lead to **logistic regression** (which is classification despite its name).

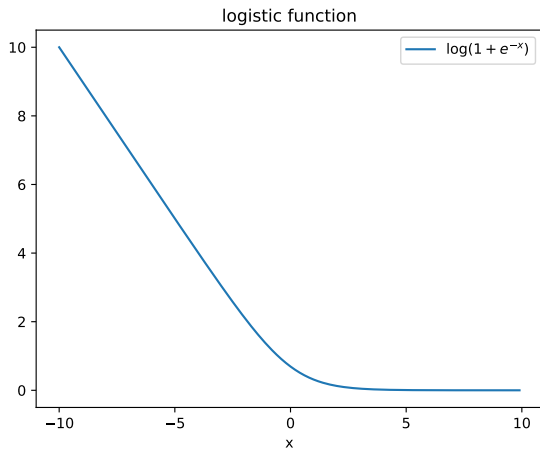
Most common convex surrogates

If $\mathcal{Y} = \{0, 1\}$, \hat{y} is the prediction and y is the correct label, then we sometimes write :

$$l(\hat{y}, y) = y \log(1 + e^{-\hat{y}}) + (1 - y) \log(1 + e^{\hat{y}}) \quad (5)$$

(cross entropy loss)

Logistic function



Most common convex surrogates

Définition

Hinge loss

$$\Phi(u) = \max(1 - u, 0) \quad (6)$$

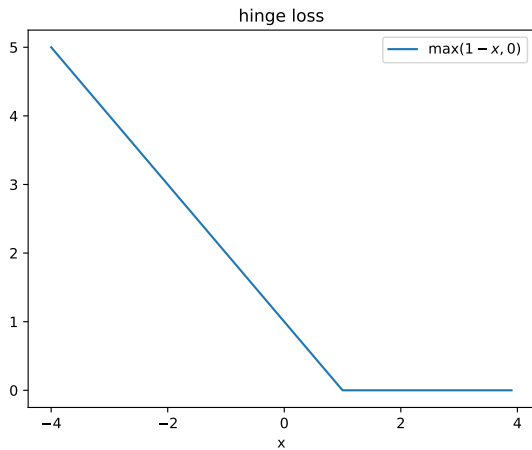
With linear predictors, this loss will lead to **Support vector machines**.

Définition

Squared hinge loss

$$\Phi(u) = (\max(1 - u, 0))^2 \quad (7)$$

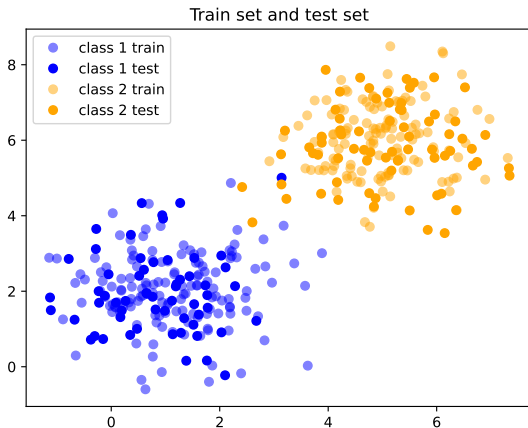
Hinge loss



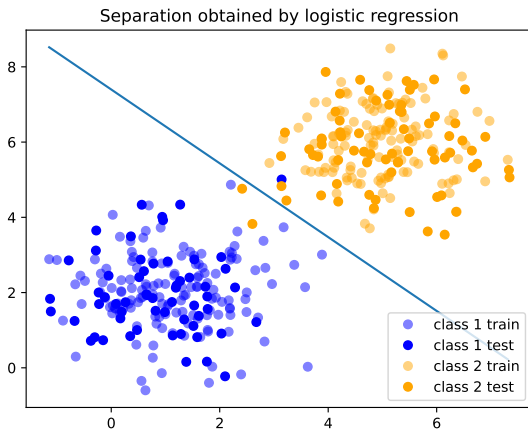
Logistic regression

- ▶ $g(x) = \langle x, \theta \rangle = x^T \theta.$
- ▶ $f(x) = \text{sign}(\langle x^T \theta \rangle)$
- ▶ It can be seen as "linear regression applied to classification".

Application on a toy dataset



Application on a toy dataset



No closed-form solution

Since the loss is convex, to minimize it is sufficient to look for the cancellation of the gradient. However, the corresponding equation has no closed-form solution.

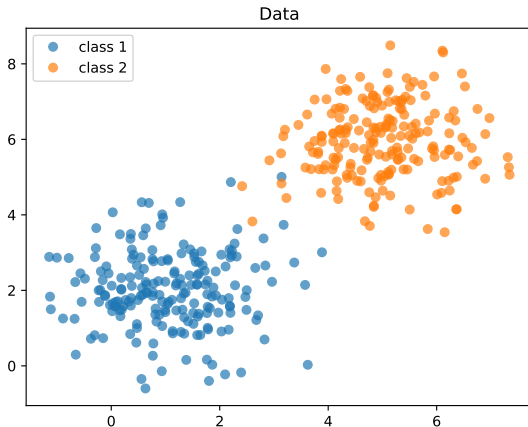
We thus need to use iterative algorithms (Gradient descent, Newton's method)

Practical usage of logistic regression

In practice, it is common practice to :

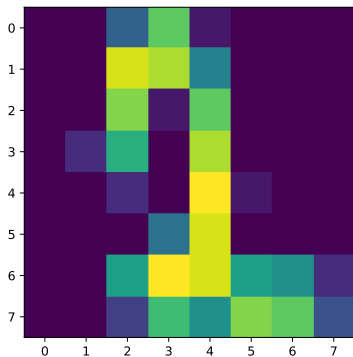
- ▶ regularize the logistic loss to avoid overfitting, for instance with a $L2$ penalty (as in ridge regression)
- ▶ use feature maps and classify with $\phi(x)$ instead of x .

Application to a simple 2D dataset.



Application to the digits dataset.

https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html



Data transformation (feature maps)

