

MACHINE LEARNING

LOGISTIC REGRESSION

Sebastian Engelke

MASTER IN BUSINESS ANALYTICS



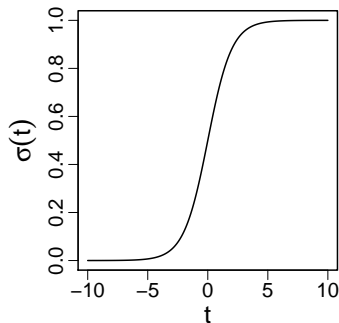
**UNIVERSITÉ
DE GENÈVE**

Logistic regression

- ▶ For simplicity we first consider classification for 2 classes $Y \in \{0, 1\}$.
- ▶ We assume the feature vector X is such that $X_1 = 1$ (intercept), so that we can simply write $x^\top \beta$ instead of $\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$.
- ▶ Logistic regression links the probability of class 1 to $x^\top \beta$ by

$$\Pr(Y = 1 \mid X = x) = \sigma(x^\top \beta) \quad \text{where} \quad \sigma(t) = \frac{\exp(t)}{1 + \exp(t)}.$$

- ▶ The function σ is called the **logistic function**: it maps \mathbb{R} to $(0, 1)$.



Logistic regression: interpretation

- We observe

$$\Pr(Y = 1 \mid X = x) = \sigma(x^\top \beta) \Leftrightarrow \sigma^{-1}(\Pr(Y = 1 \mid X = x)) = x^\top \beta,$$

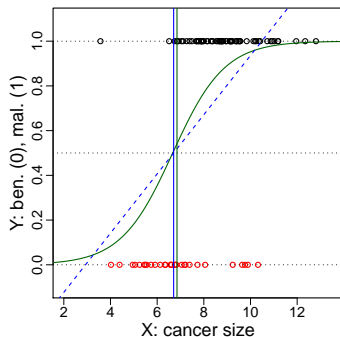
where $\sigma^{-1}(u) = \log\{u/(1-u)\}$ called the **logit function**.

- Logistic regression thus implies that **log-odds** are linear

$$\log \frac{\Pr(Y = 1 \mid X = x)}{\Pr(Y = 0 \mid X = x)} = \log \frac{\sigma(x^\top \beta)}{1 - \sigma(x^\top \beta)} = x^\top \beta,$$

which helps interpreting the coefficients β_j .

- Note that this implies that the decision boundary from logistic regression is linear.
- Figure: Logistic regression (green) versus linear regression (blue).

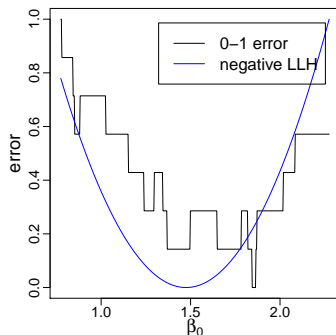


Logistic regression: estimation

- ▶ We estimate the parameter β by maximizing the **conditional log-likelihood** of the sample $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i \in \{0, 1\}$.
- ▶ By the logistic model assumption $Y \mid X = x_i$ is **Bernoulli** distributed with success probability $\sigma(x_i^\top \beta)$. This gives the log-likelihood for the sample

$$\ell(\beta) = \sum_{i=1}^n \log \Pr(Y = y_i \mid X = x_i) = \sum_{i=1}^n y_i x_i^\top \beta - \log(1 + e^{x_i^\top \beta}).$$

- ▶ The **maximum likelihood estimator** (MLE) $\hat{\beta} = \operatorname{argmax}_{\beta} \ell(\beta)$ has no explicit formula; ℓ must be maximized numerically (Newton–Raphson, gradient descent, etc.).



Logistic regression: estimation

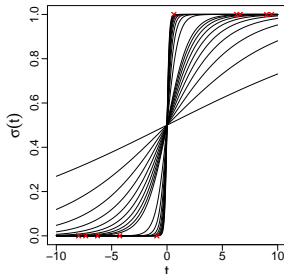
- **Prediction:** From $\hat{\beta}$ we estimate for a new input $x_0 \in \mathbb{R}^p$

$$\widehat{\Pr}(Y = 1 \mid X = x_0) = \sigma(x_0^\top \hat{\beta}).$$

We use the **Bayes classifier** based on these estimates to predict the class

$$\hat{y}_0 = \begin{cases} 1 & \text{if } \sigma(x_0^\top \hat{\beta}) > 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

- Careful: The MLE $\hat{\beta}$ **does not exist** when the classes are **perfectly separable**.
- The figure illustrate the problem of **perfect separation**. The observations are in red.
- The two classes can be perfectly separated by letting $\beta_1 \rightarrow \infty$.



Multiclass logistic regression: multinomial regression

- ▶ Logistic regression extends to the **multiclass** case $Y \in \{1, \dots, q\}$ by specifying

$$\Pr(Y = j \mid X = x) = \frac{e^{x^\top \beta^{(j)}}}{1 + \sum_{l=1}^{q-1} e^{x^\top \beta^{(l)}}}, \quad j = 1, \dots, q-1,$$
$$\Pr(Y = q \mid X = x) = \frac{1}{1 + \sum_{l=1}^{q-1} e^{x^\top \beta^{(l)}}},$$

where $\beta^{(1)}, \dots, \beta^{(q-1)} \in \mathbb{R}^p$ are parameter vectors to be estimated.

- ▶ The model uses $q - 1$ equations for q classes: this is because of the constraint that the probabilities must sum to one.
- ▶ Maximum likelihood estimation: the distribution of $Y \mid X$ is **multinomial** and we can obtain the likelihood

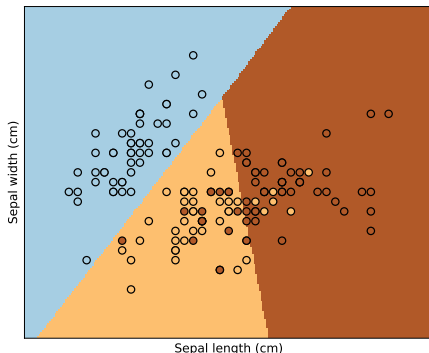
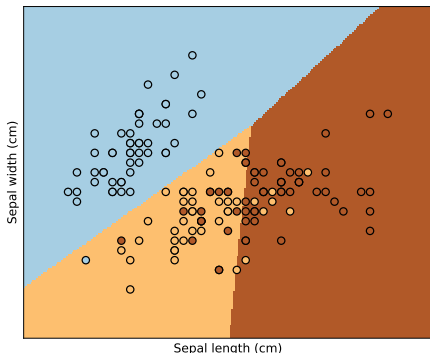
$$\ell(\beta) = \sum_{i=1}^n \log \Pr(Y = y_i \mid X = x_i).$$

Iris data set: multinomial (logistic) regression

For 150 measured plants; $p = 4$ predictors as response the three Iris species

$\mathcal{G} = \{\text{setosa}, \text{versicolor}, \text{virginica}\}$ the plant belongs. We only use two predictors here.

```
from sklearn.linear_model import LogisticRegression
iris = sklearn.datasets.load_iris()
X, y = iris.data[:, :2], iris.target # we only keep the first two features: 'Sepal length', 'Sepal width'
logreg = LogisticRegression(penalty="none")
logreg.fit(X, y)
x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
h = .02 # step size in the mesh
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z = logreg.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)
```



- ▶ Other link functions than σ can be used to map responses $x^\top \beta$ to probabilities, e.g., the so-called **probit regression**

$$\Pr(Y = 1 \mid X = x) = \Phi(x^\top \beta),$$

where Φ is the univariate Gaussian distribution function. Logistic is usually preferable to probit regression: it is more robust to outliers.

- ▶ Logistic regression is a special case of a **generalized linear model** (GLM).

Logistic regression vs. LDA

- ▶ For both models the log-odds between any two classes are linear in x .
- ▶ The difference between LDA and logistic regression comes from the **estimation** of their parameters.
- ▶ LDA models the joint distribution of (Y, X) , while logistic regression only models $Y \mid X$. If the Gaussian assumption for the conditional class densities is reasonable then LDA may perform better. If the Gaussian assumption is clearly unreasonable then logistic regression will outperform LDA.
- ▶ Often they have similar performance.