

Logistic Regression

Prof. Alessandro Lucantonio

Aarhus University - Department of Mechanical and Production Engineering

?/?/2023

Binary classification

Classification : discrete target (output vector).

Binary classification: $\{0, 1\}$ target

Example of binary classification task: Spam/not spam emails.

Idea: consider hypothesis $h_{\mathbf{w}}$ such that

$$0 \leq h_{\mathbf{w}} \leq 1.$$

- ▶ if $h_{\mathbf{w}}(\mathbf{x}) \geq 0.5$, predict 1;
- ▶ if $h_{\mathbf{w}}(\mathbf{x}) < 0.5$, predict 0.

Logistic Regression

Hypothesis: $h_{\mathbf{w}}(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x})$, where

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

is the **sigmoid function**.

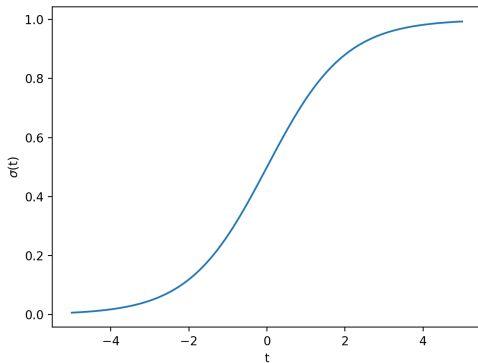


Figure: Sigmoid function

Linear decision boundary

$$\text{Model: } h_{\mathbf{w}}(x_1, x_2) = \sigma(w_0 + w_1x_1 + w_2x_2)$$

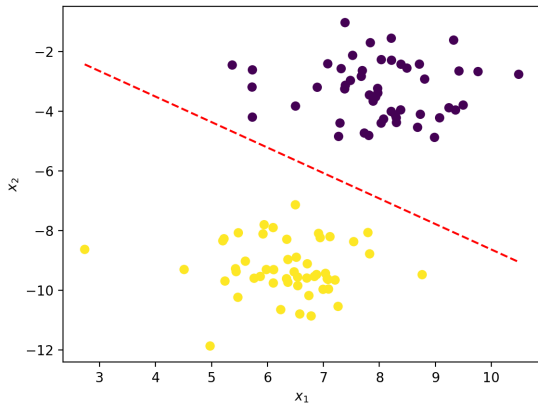


Figure: An example of linear decision boundary

Non-linear decision boundary

Model: $h_{\mathbf{w}}(x_1, x_2) = \sigma(w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2)$

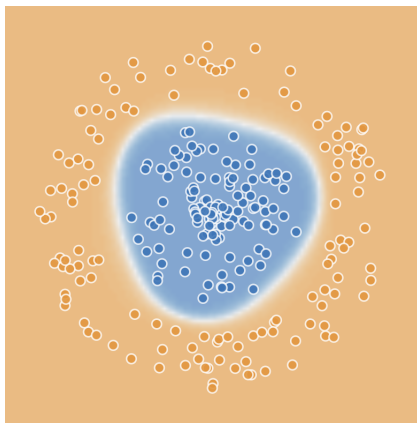


Figure: An example of non-linear decision boundary

Cost function

First try: MSE

$$E(\mathbf{w}) = \frac{1}{N} \sum_{i=0}^N (\sigma(\mathbf{w}^T \mathbf{x}^i) - y)^2$$

Huge problem: σ *non-convex*, hence MSE is *non-convex* (many local minima).

Main idea: if $y = 1$ the prediction $h_{\mathbf{w}}(x)$ is good when $h_{\mathbf{w}}(x) \approx 1$.
Good prediction means low error and $\log(1) = 0$.

Second try: **Binary cross-entropy**

$$E(\mathbf{w}) := -\frac{1}{N} \sum_{i=1}^N y^i \log(h_{\mathbf{w}}(x^i)) + (1 - y^i) \log(1 - h_{\mathbf{w}}(x^i))$$

Parenthesis - What is a convex function?

A function is *convex* when for all pairs of points on the graph, the line segment that connects these two points passes above the curve.

A function is *concave* when its opposite is convex.

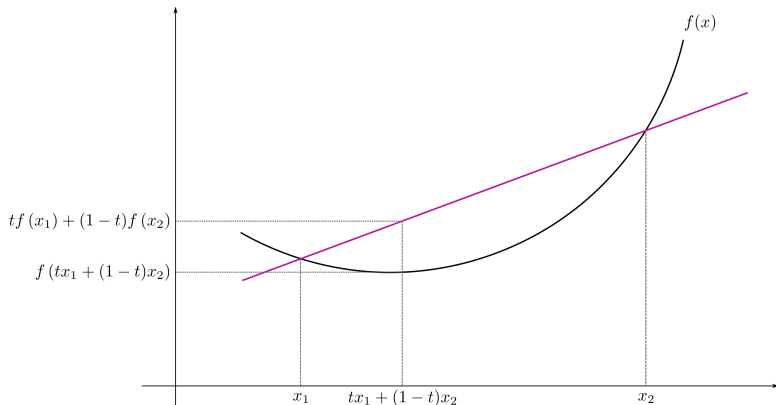


Figure: Geometric intuition of convexity

Parenthesis - The power of convexity

Any local minimum of a convex function is also a global minimum. Instead, a non-convex function has potentially many local minima which are not global minima and many saddle points (points with null gradient but nor minimum nor maximum).

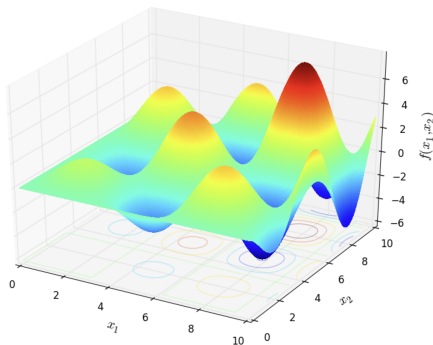


Figure: An example of non-convex function

Tips and Tricks - Is a local minimum always a bad news?

Overfitting is around the corner. Finding a global minimum means that we have the best fit possible on the training set: this is a potentially red flag on overfitting.

Of course is better to have a convex cost function, but this is not always the case. In that cases, finding a local minimum is probably even better than a global minimum.

Remember our goal: **generalization**.

Worst case scenario: find a saddle point.

Derivative of the sigmoid

Goal: Compute the gradient of the Cross-entropy.

Recall: $\sigma(t) := \frac{1}{1+e^{-t}}$.

$$\begin{aligned}\frac{d}{dt}\sigma(t) &= \frac{e^{-t}}{(1+e^{-t})^2} \\ &= \left(\frac{1}{1+e^{-t}}\right) \left(\frac{e^{-t}}{1+e^{-t}}\right) \\ &= \sigma(t) \left(1 - \frac{1}{1+e^{-t}}\right) \\ &= \sigma(t)(1 - \sigma(t)).\end{aligned}$$

Gradient of the Cross-entropy

Einstein notation: $\sum_i a_i \rightsquigarrow a_i$

$$\begin{aligned}\frac{\partial}{\partial w_j} h_{\mathbf{w}}(\mathbf{x}) &= \sigma'(\mathbf{w}^T \mathbf{x}) x_j = \sigma(\mathbf{w}^T \mathbf{x})(1 - \sigma(\mathbf{w}^T \mathbf{x})) x_j \\ &= h_{\mathbf{w}}(\mathbf{x})(1 - h_{\mathbf{w}}(\mathbf{x})) x_j\end{aligned}$$

$$\begin{aligned}N \frac{\partial}{\partial w_j} E(\mathbf{w}) &= - \left[y^i \frac{\partial}{\partial w_j} \log(h_{\mathbf{w}}(\mathbf{x}^i)) + (1 - y^i) \frac{\partial}{\partial w_j} \log(1 - h_{\mathbf{w}}(\mathbf{x}^i)) \right] \\ &= - \left[\frac{y^i \frac{\partial}{\partial w_j} h_{\mathbf{w}}(\mathbf{x}^i)}{h_{\mathbf{w}}(\mathbf{x}^i)} - \frac{(1 - y^i) \frac{\partial}{\partial w_j} (1 - h_{\mathbf{w}}(\mathbf{x}^i))}{1 - h_{\mathbf{w}}(\mathbf{x}^i)} \right]\end{aligned}$$

→

Gradient of the Cross-entropy

$$\begin{aligned} &= -[y^i(1 - h_{\mathbf{w}}(\mathbf{x}^i))x_j^i - (1 - y^i)h_{\mathbf{w}}(\mathbf{x}^i)x_j^i] \\ &= -[y^i - h_{\mathbf{w}}(\mathbf{x}^i)]x_j^i \\ &= [h_{\mathbf{w}}(\mathbf{x}^i) - y^i]x_j^i. \end{aligned}$$

Final result:

$$\frac{\partial}{\partial w_j} E(\mathbf{w}) = -\frac{1}{N} \sum_{i=0}^N [h_{\mathbf{w}}(\mathbf{x}^i) - y^i] x_j^i.$$

Vectorized version:

$$\nabla_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{N} \mathbf{X}^T (\sigma(\mathbf{X}\mathbf{w}) - \mathbf{y}).$$

Multiclass classification: One-vs-all

Multi-classification: $y \in \{0, \dots, k\}$.

Idea: solve $k + 1$ binary classification problem. Given a data sample \mathbf{x}

- ▶ Establish for any $0 \leq j \leq k$ what is the probability $h_{\mathbf{w}}^{(j)}(\mathbf{x})$ (=output of the sigmoid) that \mathbf{x} belongs to the class j .
- ▶ The final prediction will be the class which provides the maximum probability, i.e. $\arg \max_j h_{\mathbf{w}}^{(j)}(\mathbf{x})$