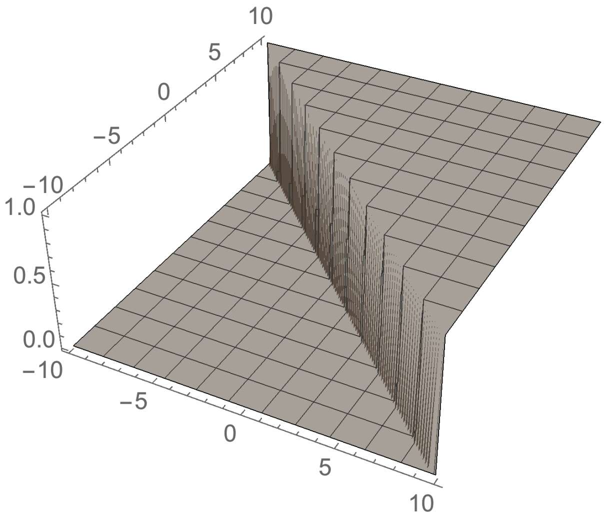
LOGISTIC REGRESSION

MOTIVATION

A classifier is a piece-wise constant function, which means it “jumps” at the decision boundary:



- We had already noted that that is inconvenient for optimization: The function is either constant (optimization algorithms cannot extract local information) or not differentiable.
- The function does not distinguish between points close to and far from the boundary. That allows e.g. the perceptron to place the decision boundary very close to data points.

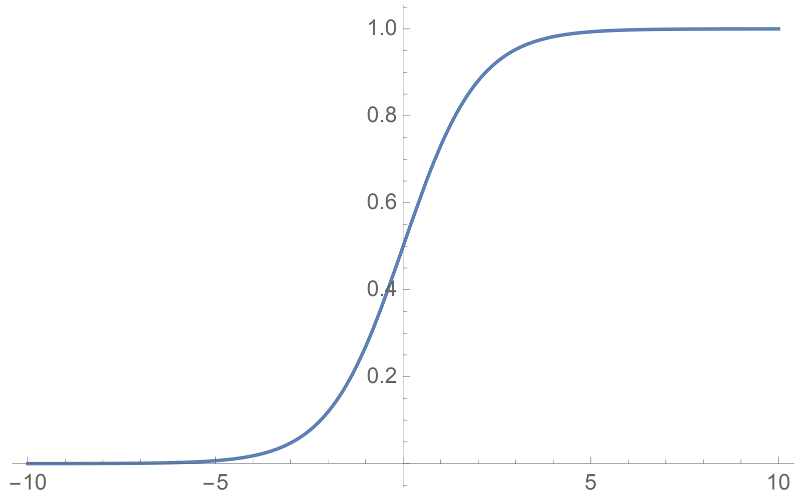
Idea

We replace the piece-wise constant function by a smooth function that otherwise looks similar. There is a canonical way of doing so, called *logistic regression*.

Keep in mind: Logistic regression is a classification method.

Sigmoid function

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

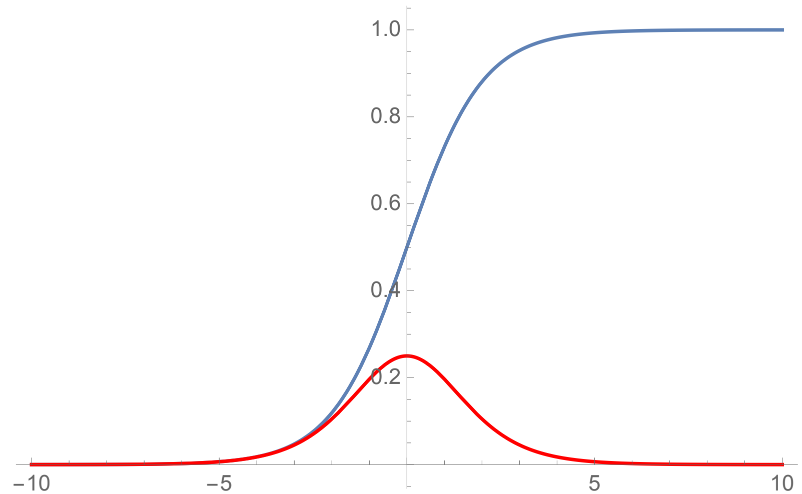


Note

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

Derivative

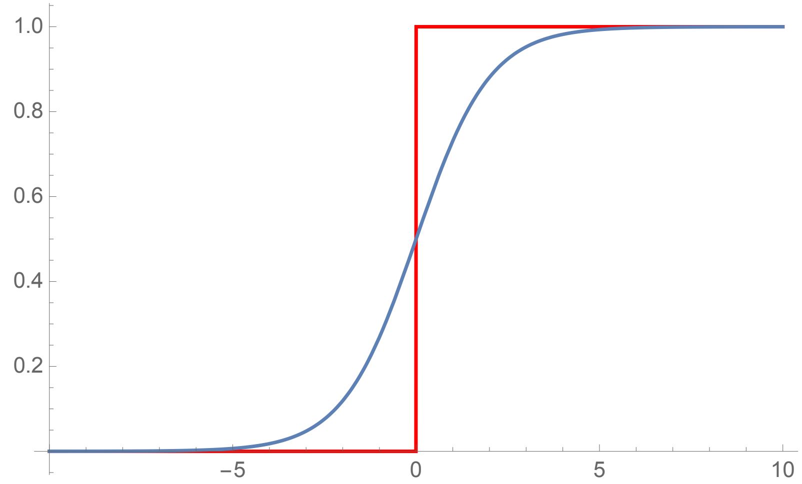
$$\frac{d\sigma}{dx}(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \sigma(x)\big(1-\sigma(x)\big)$$



Sigmoid (blue) and its derivative (red)

APPROXIMATING DECISION BOUNDARIES

- In linear classification: Decision boundary is a discontinuity
- Boundary is represented either by indicator function $\mathbb{I}\{\bullet > c\}$ or sign function $\text{sign}(\bullet - c)$
- These representations are equivalent: Note $\text{sign}(\bullet - c) = 2 \cdot \mathbb{I}\{\bullet > c\} - 1$



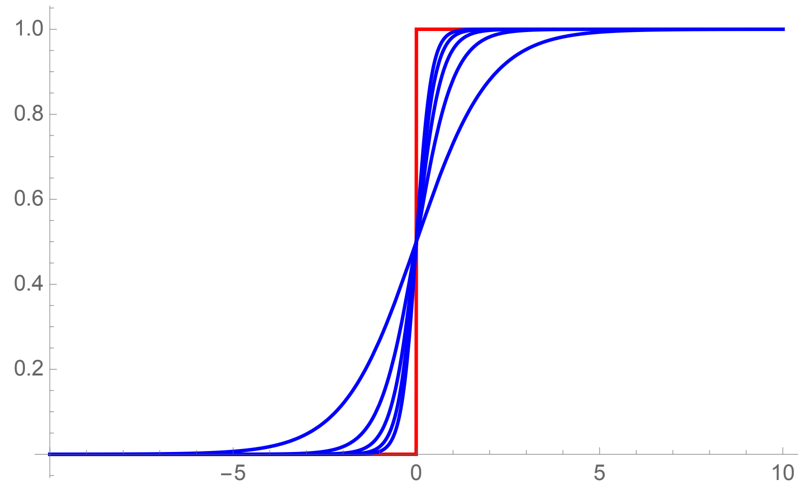
The most important use of the sigmoid function in machine learning is *as a smooth approximation to the indicator function.*

Given a sigmoid σ and a data point x, we decide which side of the approximated boundary we are own by thresholding

$$\sigma(x) \geq \frac{1}{2}$$

We can add a scale parameter by definining

$$\sigma_\theta(x) := \sigma(\theta x) = \frac{1}{1 - e^{-\theta x}} \qquad \text{for } \theta \in \mathbb{R}$$



Influence of θ

- As θ increases, σ_θ approximates $\mathbb{I}$ more closely.
- For $\theta \to \infty$, the sigmoid converges to $\mathbb{I}$ pointwise, that is: For every $x \neq 0$, we have

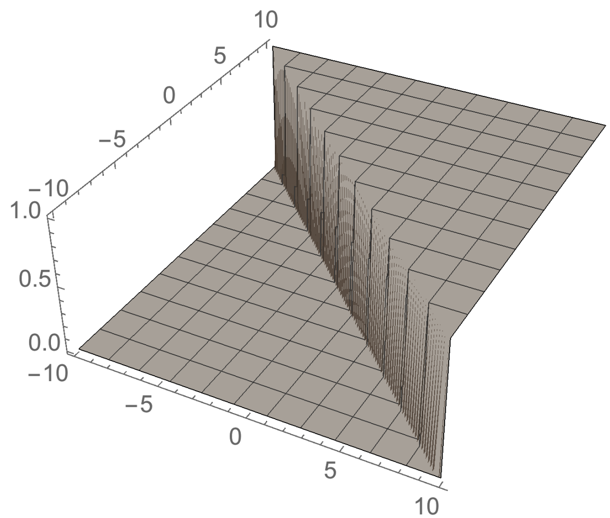
$$\sigma_\theta(x) \to \mathbb{I}\{x > 0\} \qquad \text{as } \theta \to +\infty \; .$$

- Note $\sigma_\theta(0) = \frac{1}{2}$ always, regardless of θ.

APPROXIMATING A LINEAR CLASSIFIER

So far, we have considered $\mathbb{R}$, but linear classifiers usually live in $\mathbb{R}^d$.

The decision boundary of a linear classifier in $\mathbb{R}^2$ is a discontinuous ridge:

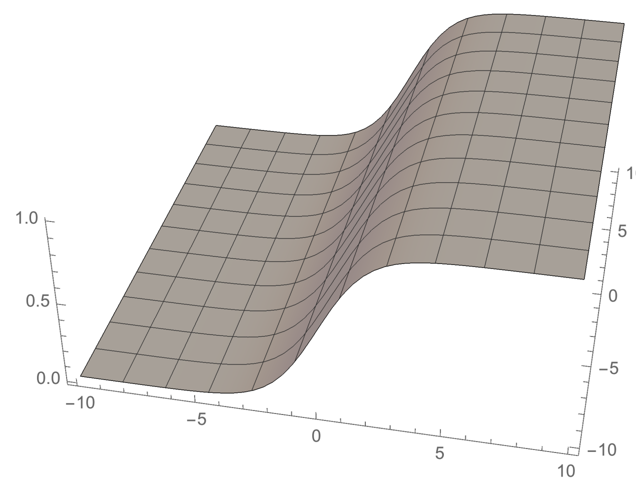


- This is a linear classifier of the form

$$\mathbb{I}\{\langle \mathbf{v}, \mathbf{x} \rangle - c\}.$$

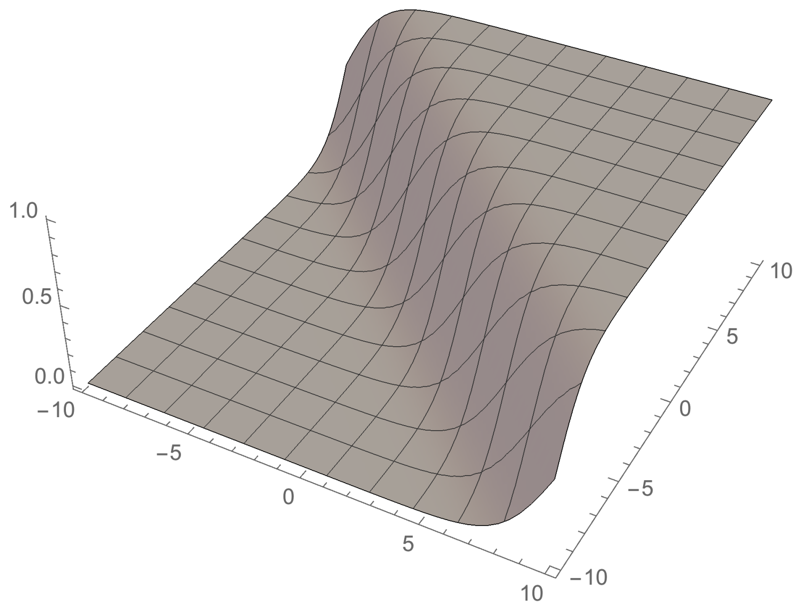
- Here: $\mathbf{v} = (1, 1)$ and $c = 0$.

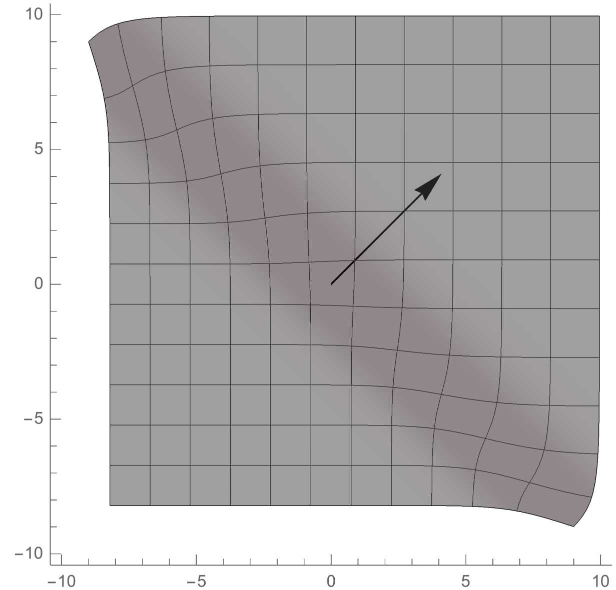
We can "stretch" σ into a ridge function on $\mathbb{R}^2$:



- This is the function $\mathbf{x} = (x_1, x_2) \mapsto \sigma(x_1)$.
- The ridge runs parallel to the x_2-axes.
- If we use $\sigma(x_2)$ instead, we rotate by 90 degrees (still axis-parallel).

Just as for a linear classifier, we use a normal vector $\mathbf{v} \in \mathbb{R}^d$.





- The function $\sigma(\langle \mathbf{v}, \mathbf{x} \rangle - c)$ is a sigmoid ridge, where the ridge is orthogonal to the normal vector $\mathbf{v}$, and c is an offset that shifts the ridge "out of the origin".
- The plot on the right shows the normal vector (here: $\mathbf{v} = (1, 1)$) in black.
- The parameters $\mathbf{v}$ and c have the same meaning for $\mathbb{I}$ and σ, that is, $\sigma(\langle \mathbf{v}, \mathbf{x} \rangle - c)$ approximates $\mathbb{I}\{\langle \mathbf{v}, \mathbf{x} \rangle \geq c\}$.

LOGISTIC REGRESSION

Logistic regression is a classification method that approximates decision boundaries by sigmoids.

Setup

- Two-class classification problem
- Observations $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$, class labels $y_i \in \{0, 1\}$.

The logistic regression model

We model the conditional distribution of the class label given the data as

$$P(y|\mathbf{x}) := \text{Bernoulli}\big(\sigma(\langle \mathbf{v}, \mathbf{x} \rangle - c)\big) \ .$$

- Recall $\sigma(\langle \mathbf{v}, \mathbf{x} \rangle - c)$ takes values in $[0, 1]$ for all θ, and value $\frac{1}{2}$ on the class boundary.
- The logistic regression model interprets this value as the probability of being in class y.

Learning Logistic Regression

Since the model is defined by a parametric distribution, we can apply maximum likelihood.

Likelihood function of the logistic regression model

$$\prod_{i=1}^{n} \sigma(\langle \mathbf{v}, \tilde{\mathbf{x}}_i \rangle - c)^{y_i} \big(1 - (\sigma(\langle \mathbf{v}, \tilde{\mathbf{x}}_i \rangle - c))\big)^{1-y_i}$$

Negative log-likelihood

$$L(\mathbf{w}) \quad := \quad -\sum_{i=1}^{n} \Big(y_i \log \sigma(\langle \mathbf{v}, \tilde{\mathbf{x}}_i \rangle - c) + (1 - y_i) \log\big(1 - \sigma((\langle \mathbf{v}, \tilde{\mathbf{x}}_i \rangle - c))\big) \Big)$$

$$\nabla L(\mathbf{v}, c) \quad = \quad \sum_{i=1}^{n} \big(\sigma(\langle \mathbf{v}, \tilde{\mathbf{x}}_i \rangle - c) - y_i\big) \begin{pmatrix} \tilde{\mathbf{x}}_i \\ 1 \end{pmatrix}$$

Note

- Each training data point $\mathbf{x}_i$ contributes to the sum proportionally to the approximation error $\sigma(\langle \mathbf{v}, \tilde{\mathbf{x}}_i \rangle - c) - y_i$ incurred at $\mathbf{x}_i$ by approximating the linear classifier by a sigmoid.

Learning logistic regression

> To learn a logistic regression classifier from training data, we minimize $L(\mathbf{v}, c)$ using gradient descent or another optimization algorithm.

- The function L is convex (= $\cup$-shaped). That means there is only a single local minimum, which is also the global minimum.
- FYI: You may encounter an algorithm called *iteratively reweighted least squares* for training logistic regression in the literature. The algorithm is obtained by applying a more sophisticated version of gradient descent (called *Newton's method*) to minimize L.