Logistic Regression

Machine Learning Course - CS-433
Oct 21, 2021
Nicolas Flammarion

EPFL

Binary classification

We observe some data $S = \{x_i, y_i\}_{i=1}^n \in \mathscr{X} \times \{0,1\}$

Goal: given a new x, we want to predict its label y

How:

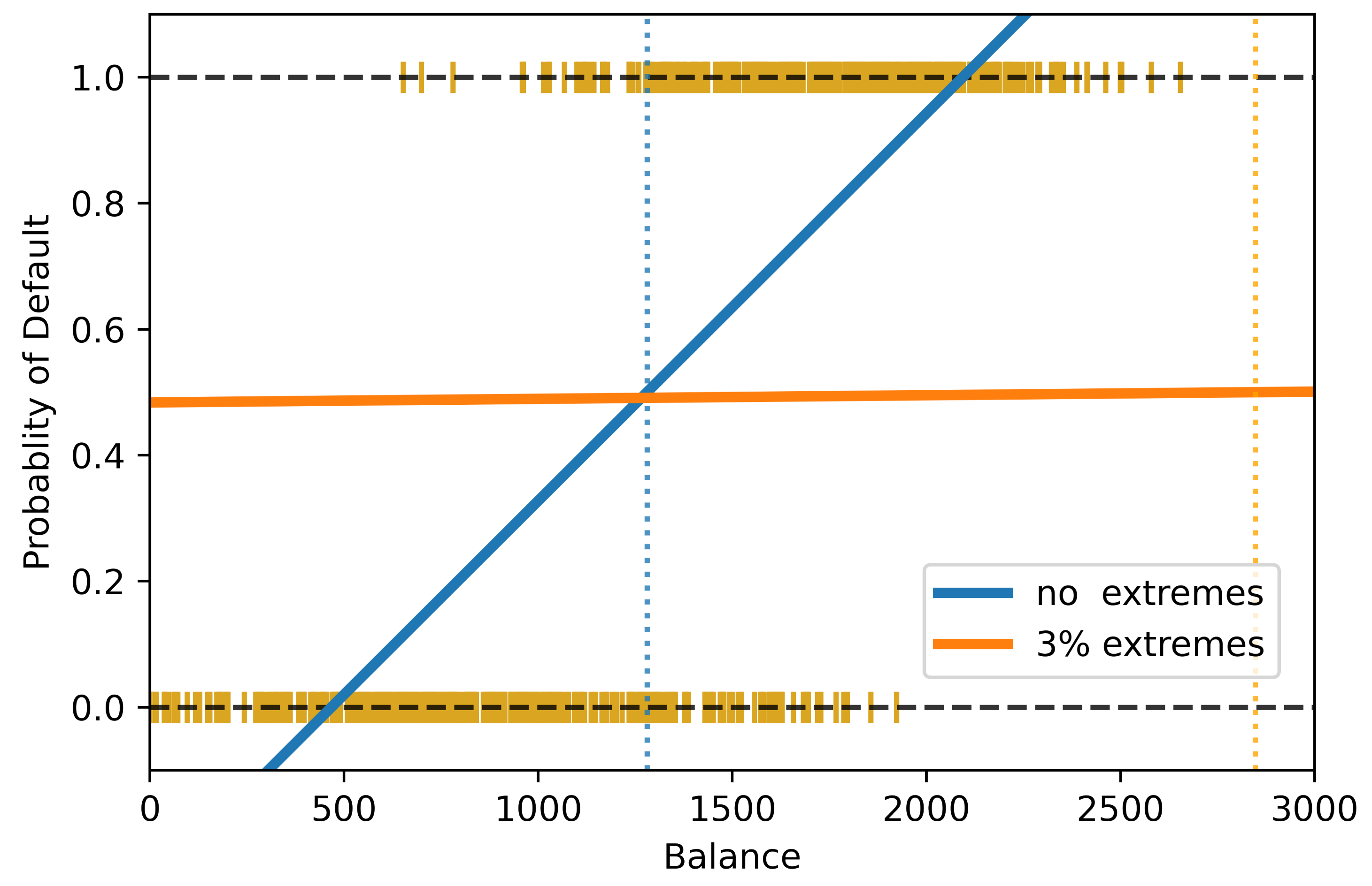
$S = \{x_i, \underset{\{0,1\}}{\underset{\cap}{y_i}}\}_{i=1}^n$ $\qquad \mathscr{A} \qquad$ $f_S = \mathscr{A}(S)$

Motivation for logistic regression

Rather than modeling the output Y directly, we can **model the probability** that Y belongs to a particular category. How?

In the previous lecture, we used a linear regression model $\mathbb{P}(Y = 1 \mid X = x) = x^\top w + w_0$ but

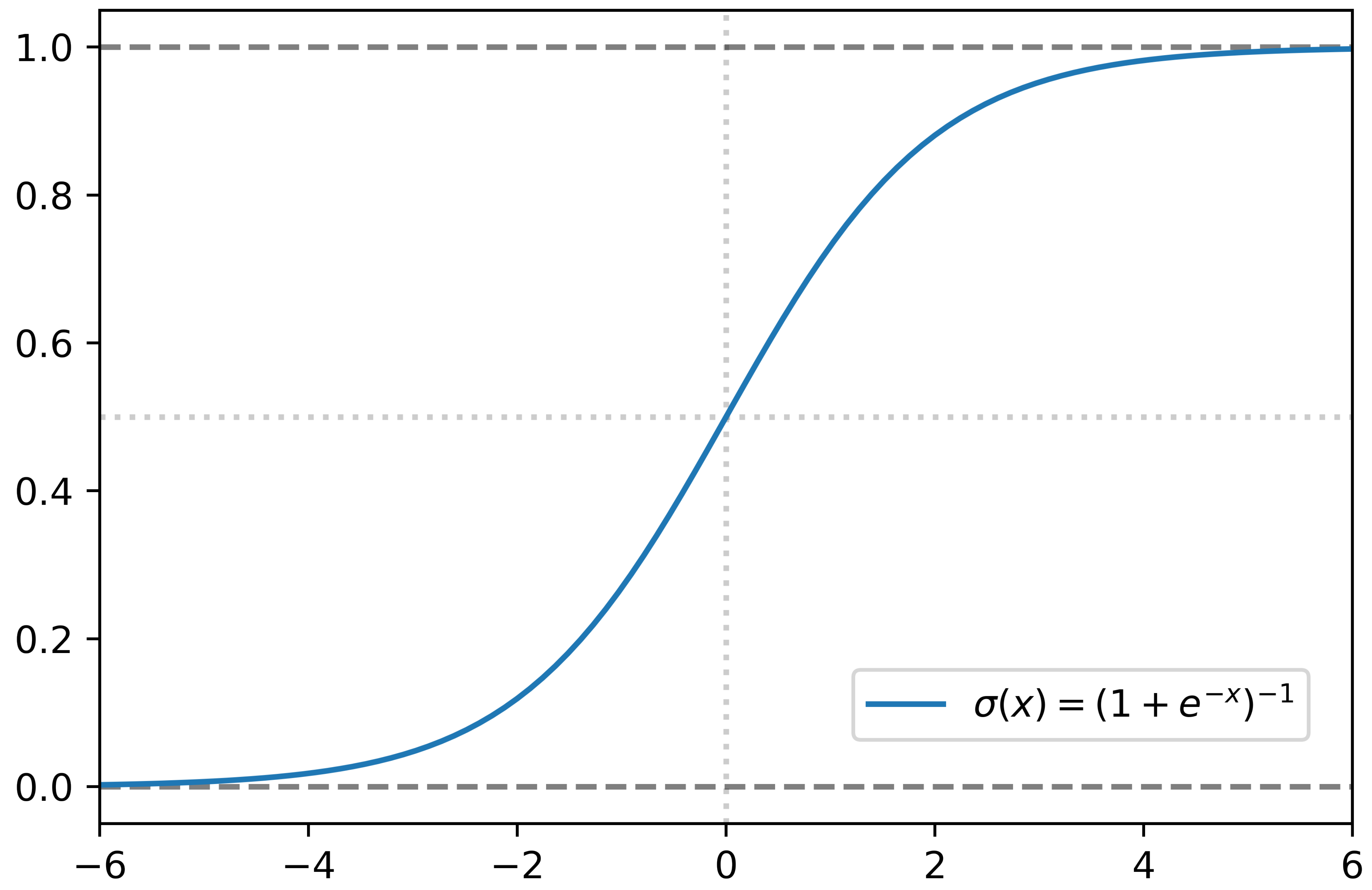
- The predicted value is not in $[0,1]$
- Very large or small values of the prediction contribute to the error even if they indicate we are very confident in the resulting classification



Solution: map the prediction from $(-\infty, +\infty)$ to $[0,1]$

The logistic function

$$\sigma(\eta) := \frac{e^\eta}{1 + e^\eta}$$



Properties of the logistic function:

- $1 - \sigma(\eta) = \dfrac{1 + e^\eta - e^\eta}{1 + e^\eta} = (1 + e^\eta)^{-1}$
- $\sigma'(\eta) = \dfrac{e^\eta(1 + e^\eta) - e^\eta e^\eta}{(1 + e^\eta)^2} = \dfrac{e^\eta}{(1 + e^\eta)^2} = \sigma(\eta)(1 - \sigma(\eta))$

Logistic Regression

$$p(1\,|\,x) := \mathbb{P}(Y = 1\,|\,X = x) = \sigma(x^\top w + w_0)$$

$$p(0\,|\,x) := \mathbb{P}(Y = 0\,|\,X = x) = 1 - \sigma(x^\top w + w_0)$$

Logistic regression models the **probability that Y belongs to a particular class** using the **logistic function σ**

Label prediction: **quantize** the probability:

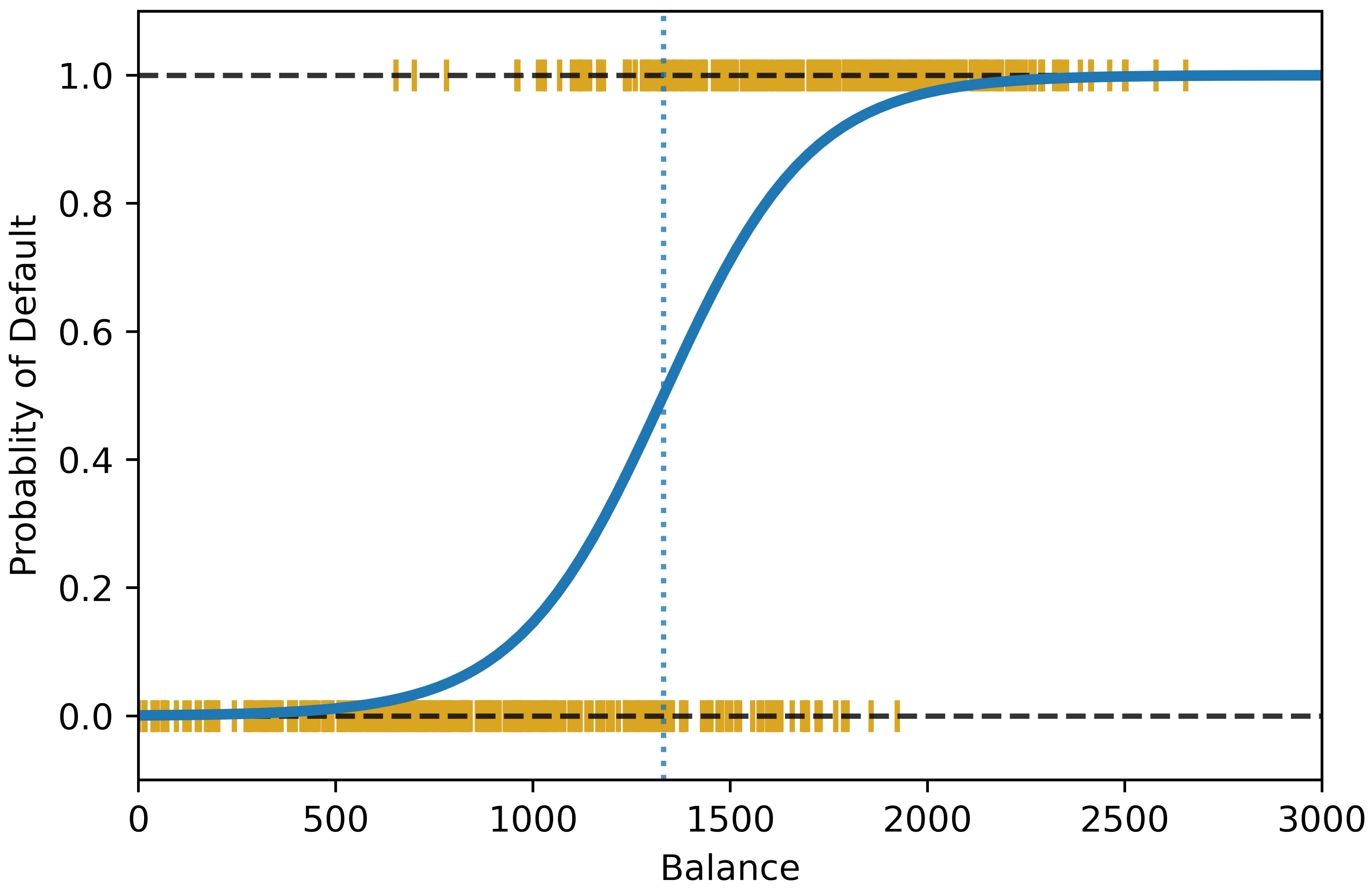
If $p(1\,|\,x) \geq 1/2$, you predict the class 1

If $p(1\,|\,x) < 1/2$, you predict the class 0

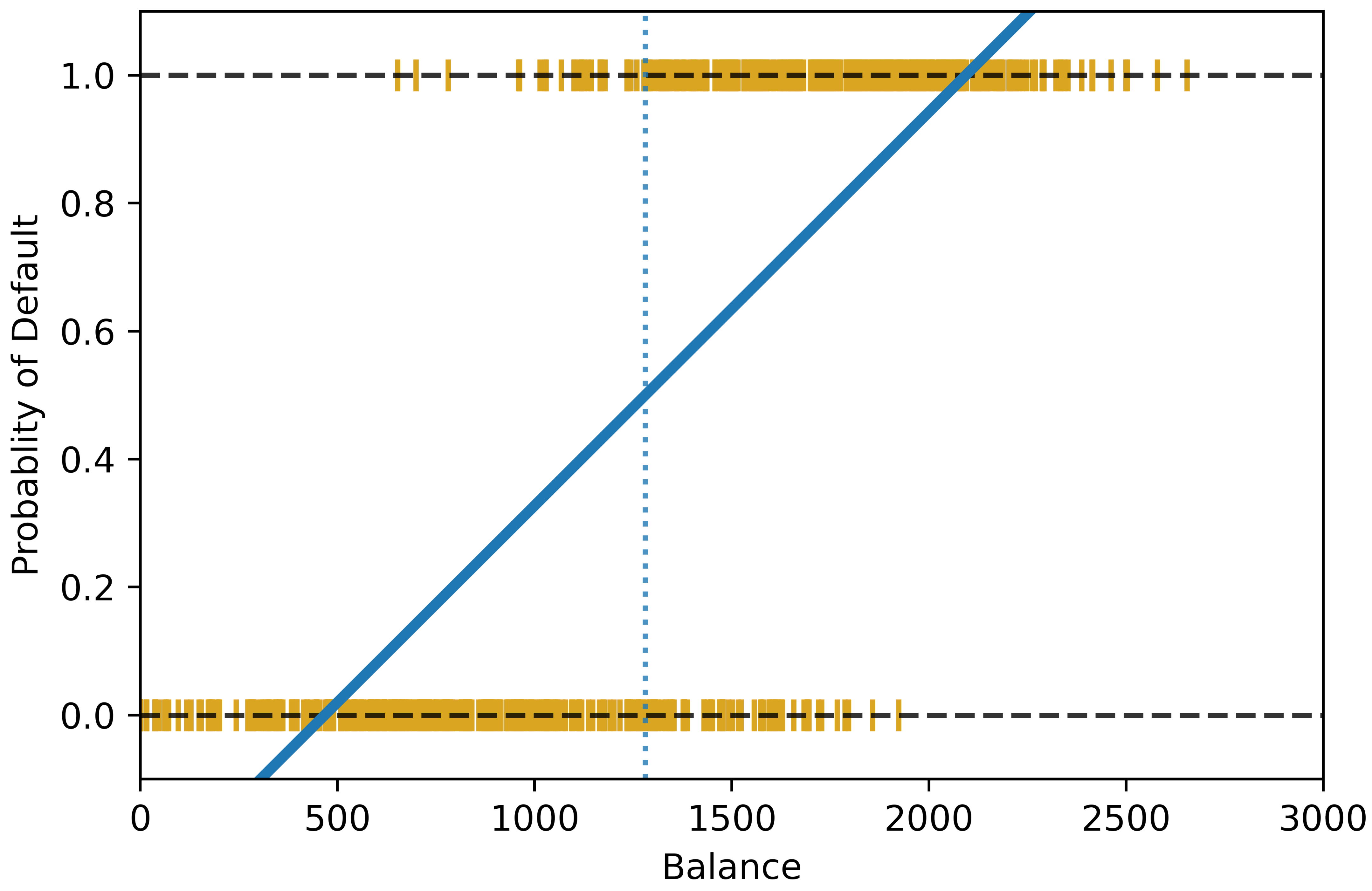
Interpretation:

- Very large $|x^\top w + w_0|$ corresponds to $p(1\,|\,x)$ very close to 0 or 1 (high confidence)
- Small $|x^\top w + w_0|$ corresponds to $p(1\,|\,x)$ very close to .5 (low confidence)

Comparison of logistic and linear regression for balanced data

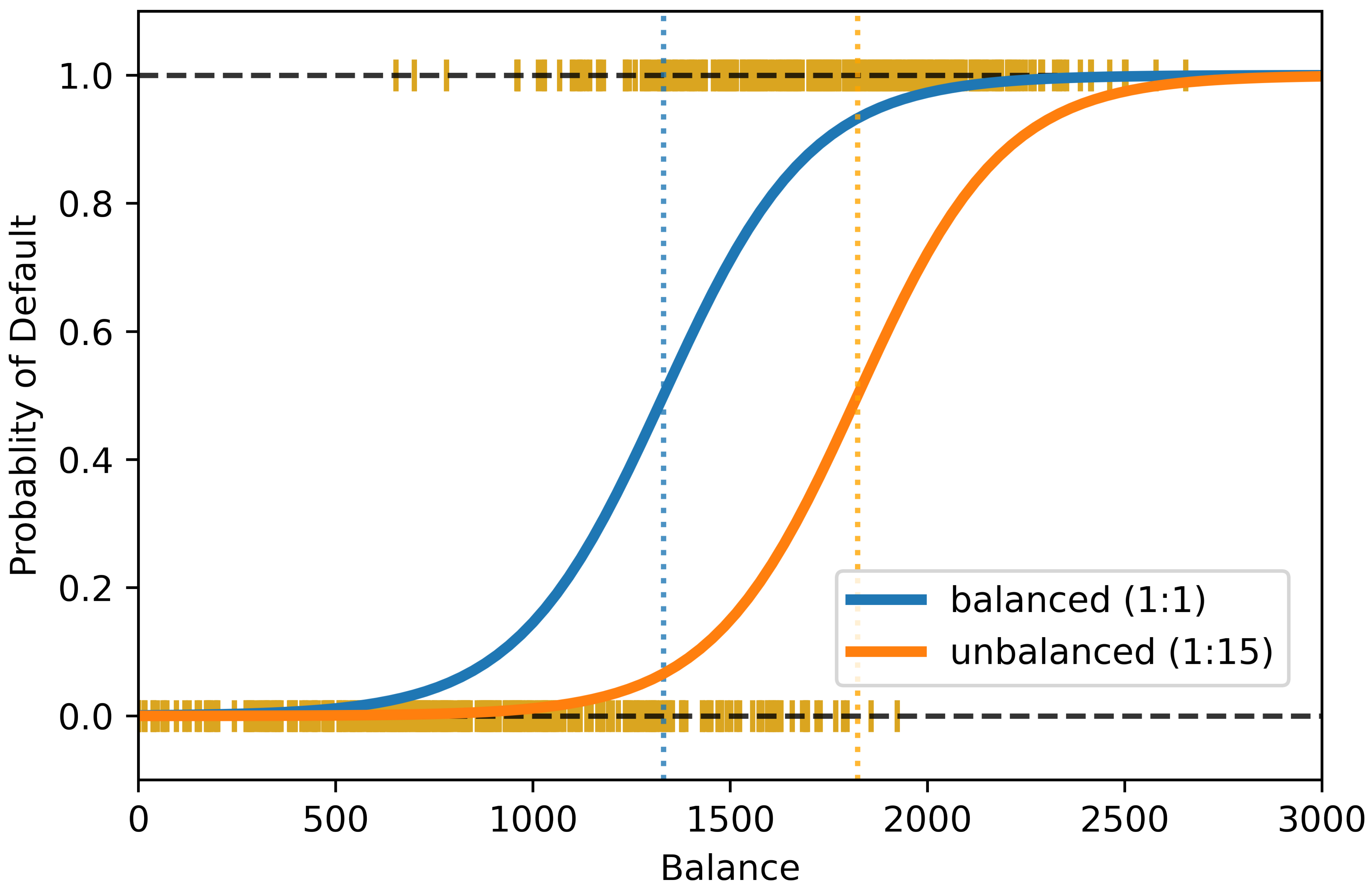


Logistic regression

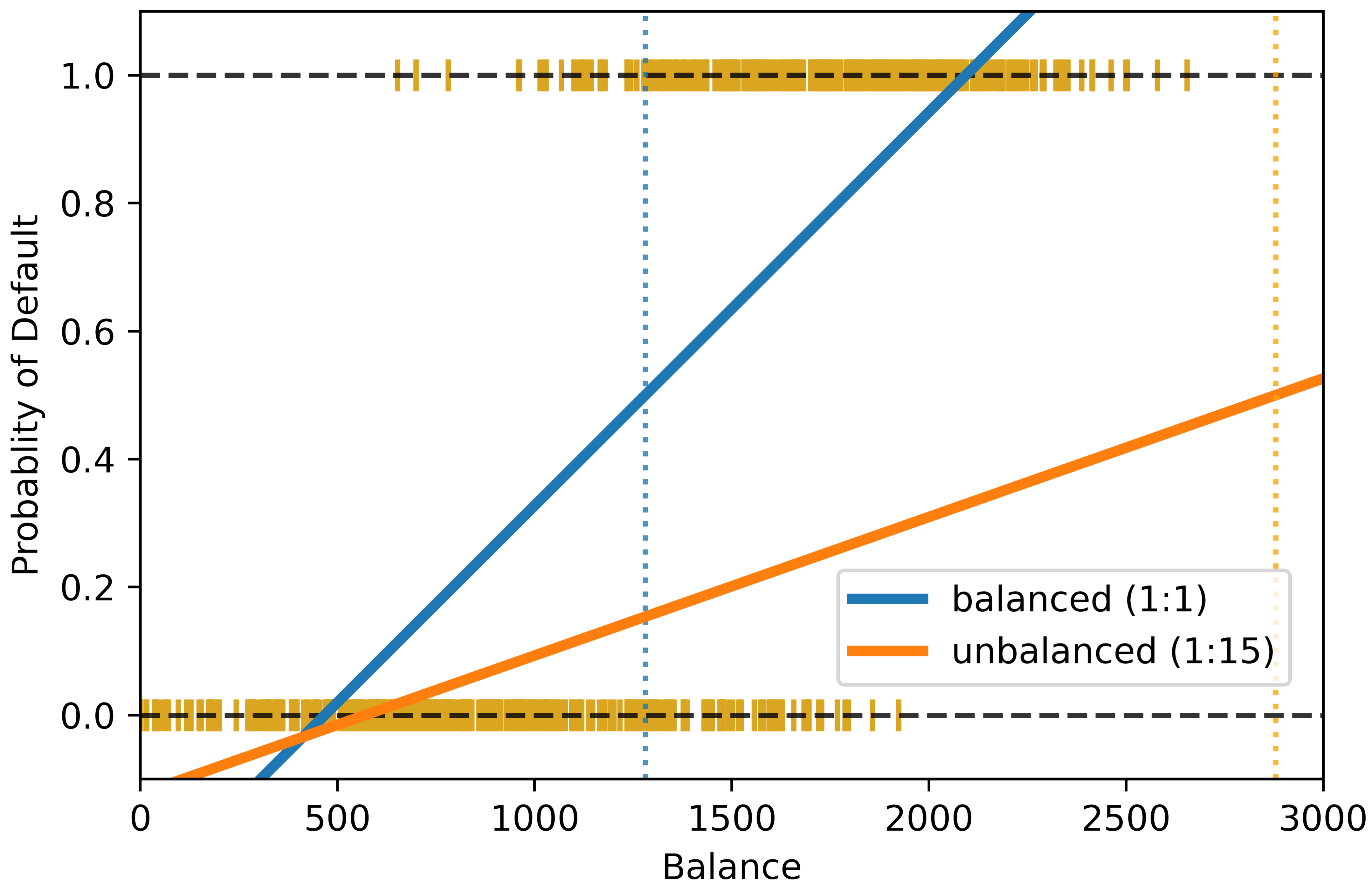


Linear regression

Comparison of logistic and linear regression for unbalanced data

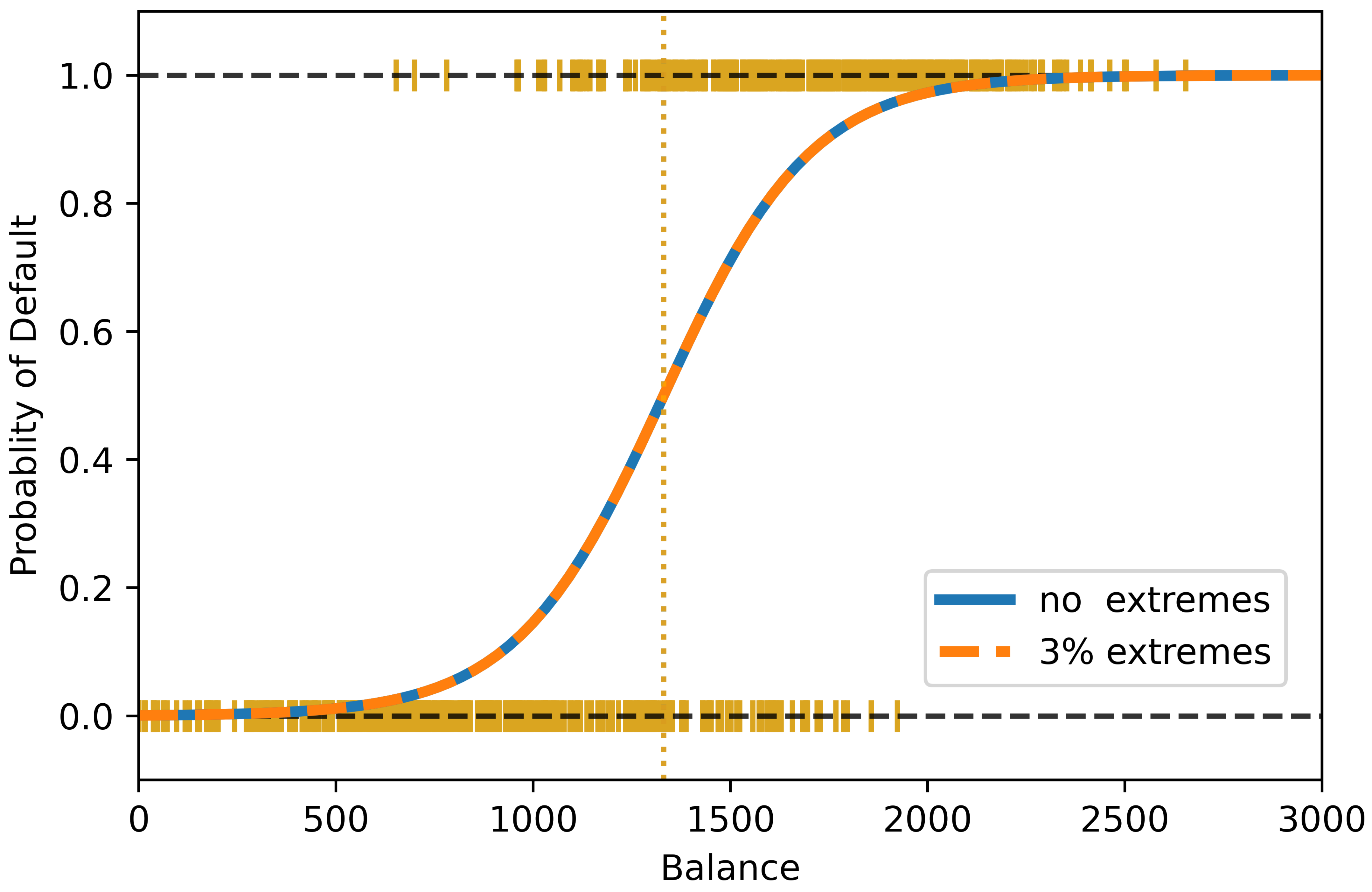


Logistic regression

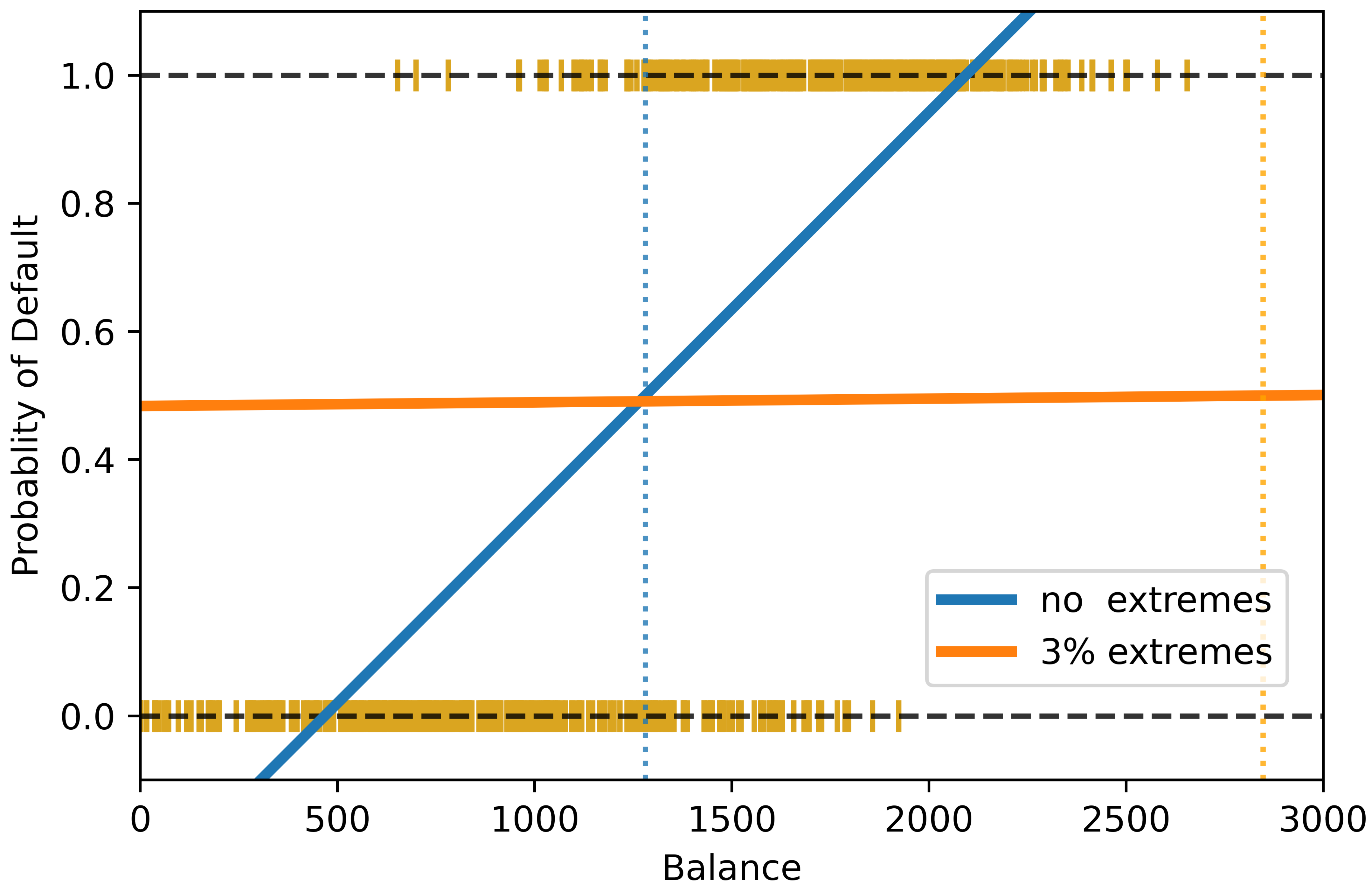


Linear regression

Comparison of logistic and linear regression for data with extreme values

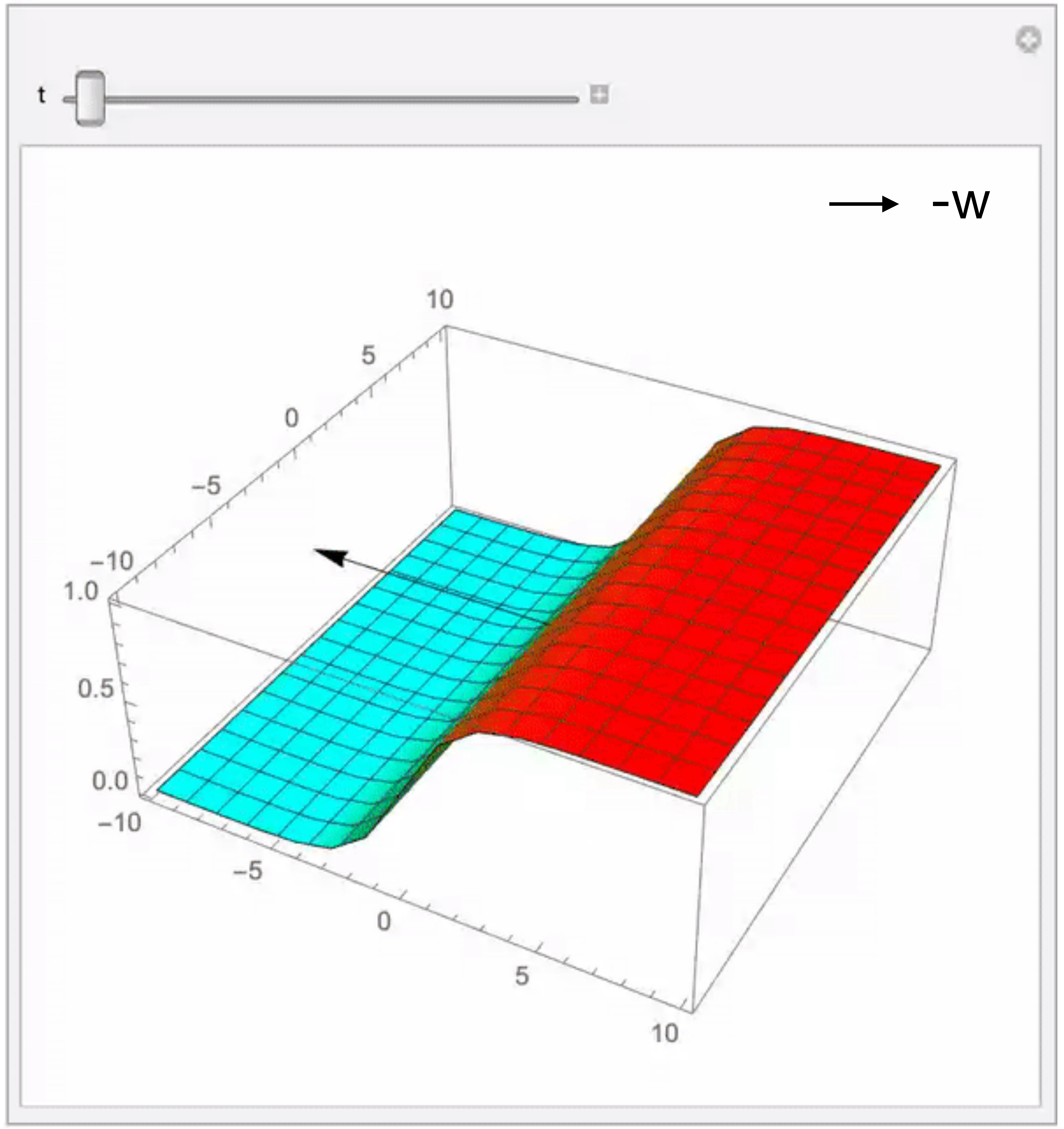


Logistic regression



Linear regression

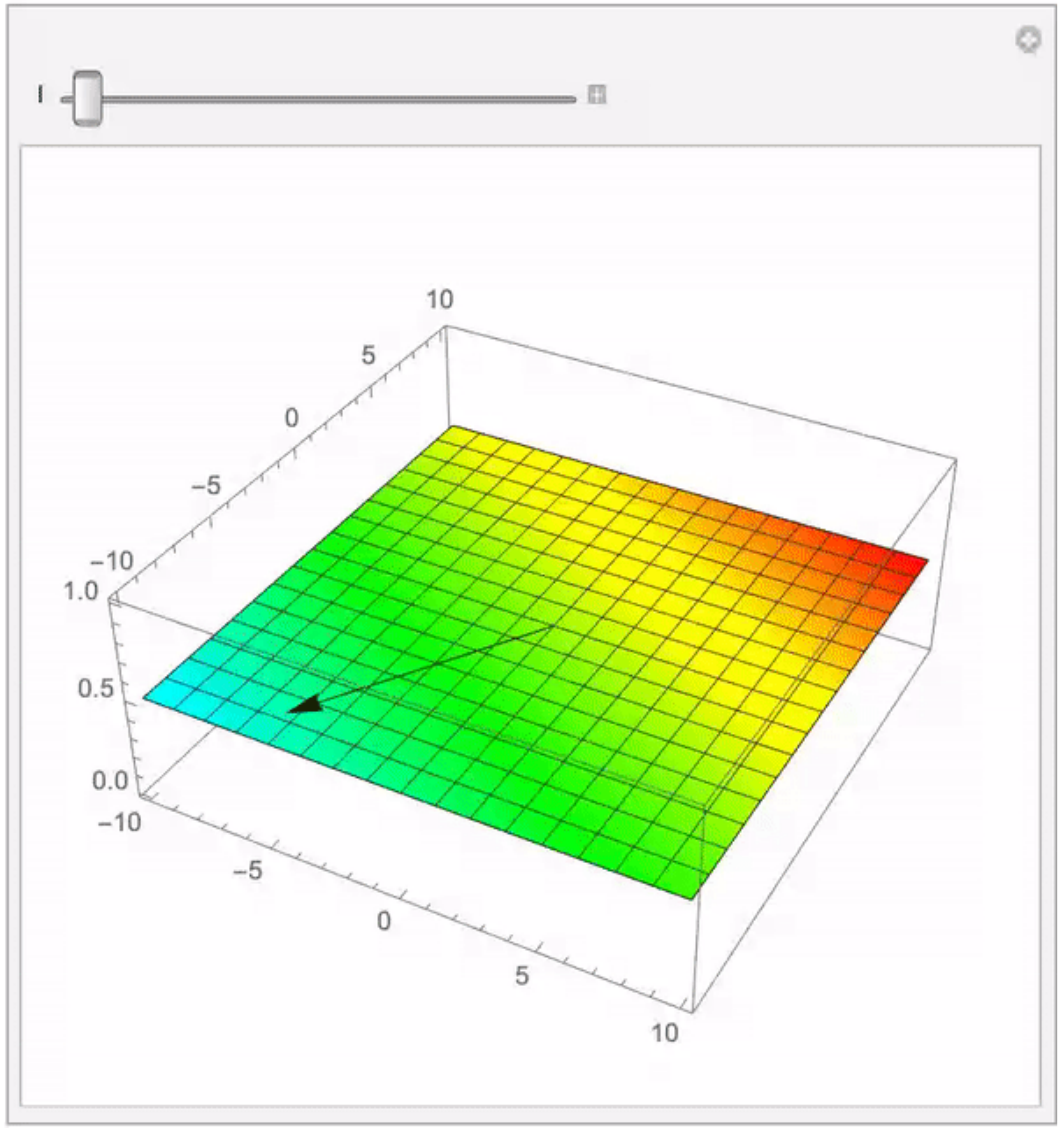
The vector w is orthogonal to the "surface of transition"



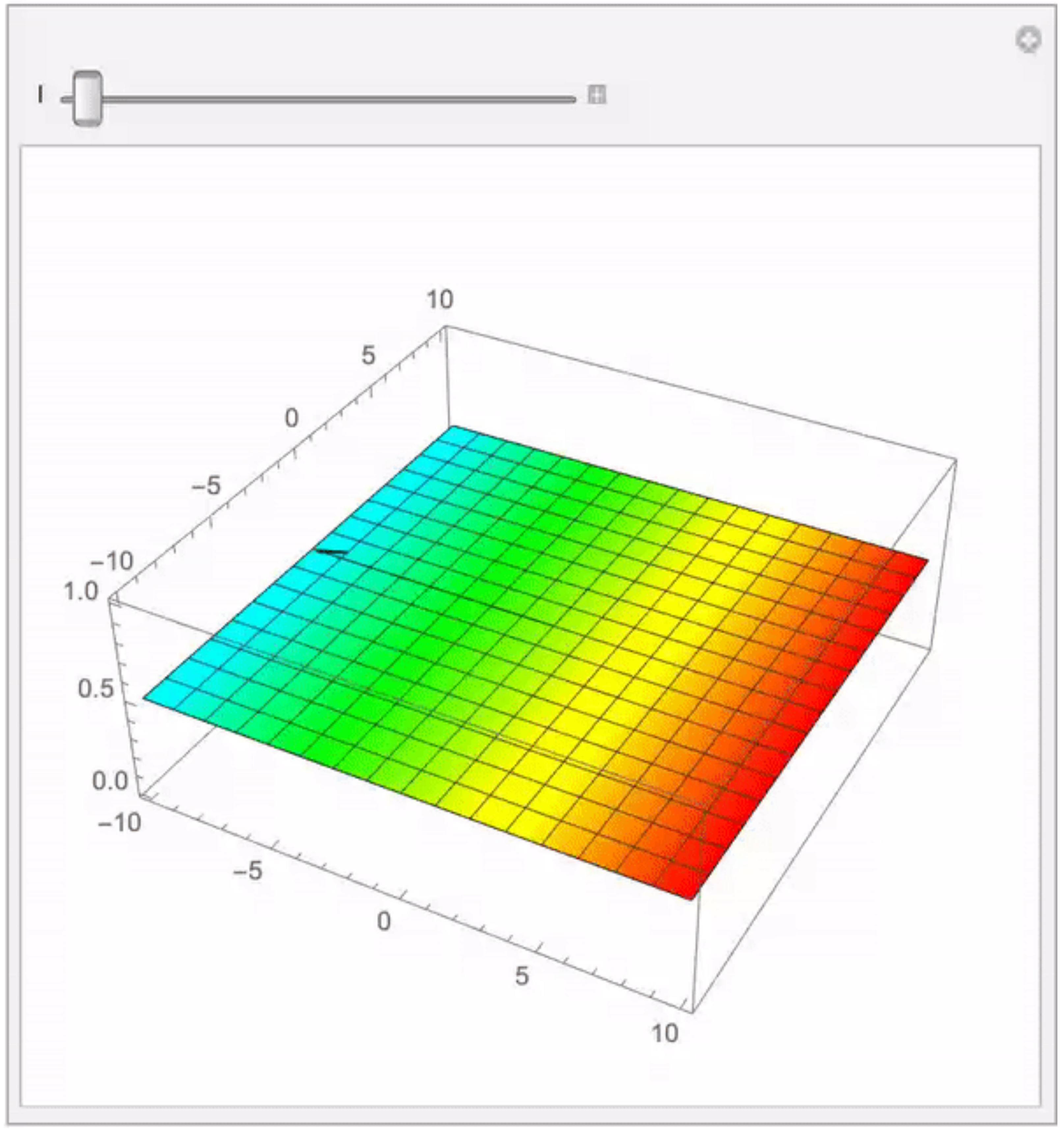
$\sigma(w^\top x)$ for $\|w\| = 1$

The transition between the two levels happens at the hyperplane $w^\perp = \{v, v^\top w = 0\}$

Scaling w makes the transition faster or slower

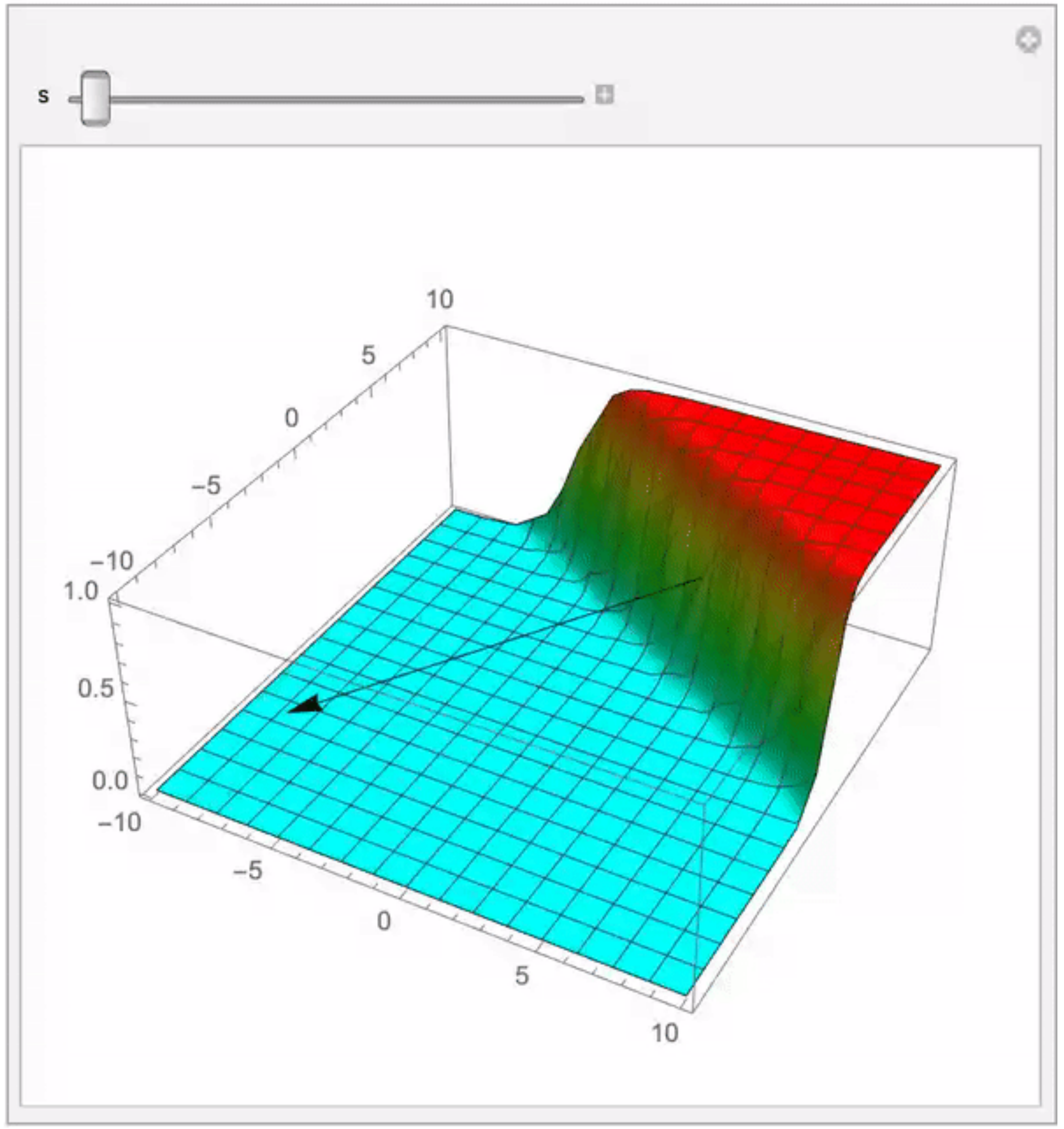


$\sigma(t \cdot w_1^\top x)$ for $t \in [e^{-10}, e^{10}]$

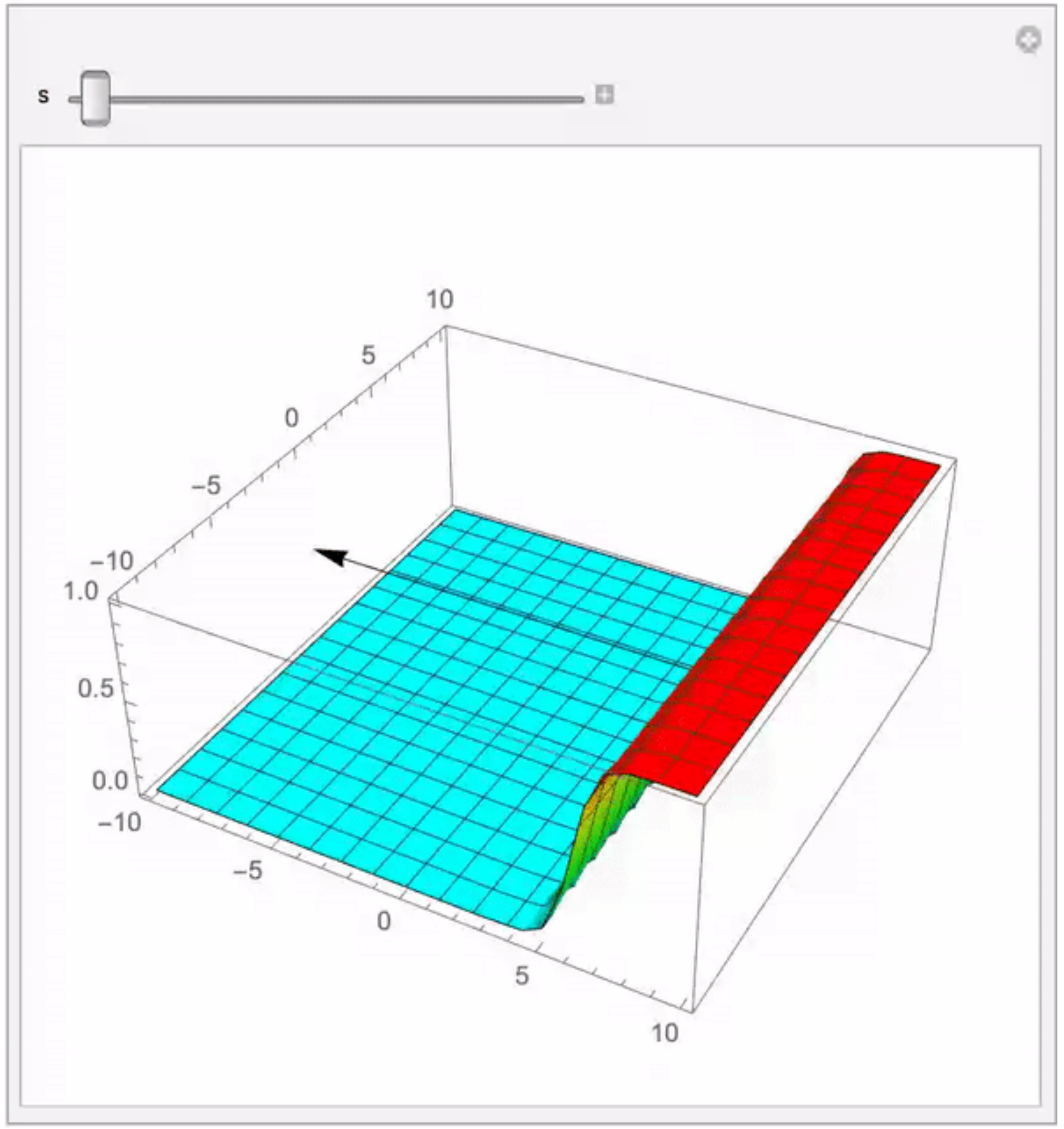


$\sigma(t \cdot w_2^\top x)$ for $t \in [e^{-10}, e^{10}]$

Changing w_0 shifts the decision region along the w vector



$\sigma(w_1^\top x + w_0)$ for $w_0 \in [-6,6]$



$\sigma(w_2^\top x + w_0)$ for $w_0 \in [-6,6]$

The transition happens at the hyperplane $\{v, v^\top w + w_0 = 0\}$

What about the bias term?

We should consider a **shift** w_0 since there is no reason that the transition hyperplan stops by 0:

$$p(1 \mid x) = \sigma(w^\top x + w_0)$$

However, for simplicity, we will prefer to **add the constant** 1 to the feature vector

$$x = \begin{pmatrix} x \\ 1 \end{pmatrix}$$

It is crucial for allowing to shift the decision region.

Note that **both options are equivalent**

Maximum likelihood estimation (MLE) is a method of estimating the parameter of a statistical model

Given i.i.d. samples $(z_1, \cdots, z_n) \sim p(z_1, \cdots, z_n, w)$, the MLE finds the **parameter** w_* **under which the observation** $z_1, \cdots, z_n$ **are the most likely**:

$$w_* = \arg\max \underset{\nearrow}{\mathscr{L}}(w) := p(z_1, \cdots, z_n, w) \underset{\nearrow}{=} \prod_{i=1}^{n} p(z_i, w)$$

Likelihood function **i.i.d. obs**

Often more convenient to work with the **negative log-likelihood**:

$$w_* = \arg\min L(w) := -\log(\mathscr{L}(w)) = -\sum_{i=1}^{n} \log(p(z_i, w))$$

This estimator is **consistent***: if the data are generated according to the model, the MLE converges to the true parameter when $n \to \infty$

In practice, data are not generated according to it, but it still provides a theoretical justification

*under mild technical conditions

MLE for logistic regression

<u>Assumption:</u> The inputs $\mathbf{X}$ does not depend on the parameter w we choose:

$$\mathcal{L}(w) = p(\mathbf{y}, \mathbf{X} \,|\, w) = p(\mathbf{X} \,|\, w) p(\mathbf{y} \,|\, \mathbf{X}, w) \underset{\mathbf{X} \perp\!\!\!\perp w}{=} p(\mathbf{X})\; p(\mathbf{y} \,|\, \mathbf{X}, w)$$

cst independant of w

$$\begin{aligned} p(\mathbf{y} \,|\, \mathbf{X}, w) &= \Pi_{i=1}^{n}\; p(y_i \,|\, x_i, w) \\ &= \Pi_{i:y_i=1}\; p(y_i = 1 \,|\, x_i, w)\; \Pi_{i:y_i=0}\; p(y_i = 0 \,|\, x_i, w) \\ &= \Pi_{i=1}^{n}\; \sigma(x_i^\top w)^{y_i} [1 - \sigma(x_i^\top w)]^{1-y_i} \end{aligned}$$

The likelihood is proportional to:

$$\boxed{\mathcal{L}(w) \propto \prod_{i=1}^{n} \sigma(x_i^\top w)^{y_i} [1 - \sigma(x_i^\top w)]^{1-y_i}}$$

Minimum of the negative log likelihood

It is more convenient to work with the negative log-likelihood:

$$
\begin{aligned}
-\log\left(p(\mathbf{y}\,|\,\mathbf{X}, w)\right) &= -\log\left(\prod_{i=1}^{n} \sigma(x_i^\top w)^{y_i}[1-\sigma(x_i^\top w)]^{1-y_i}\right) \\
&= -\sum_{i=1}^{n} y_i \log \sigma(x_i^\top w) + (1-y_i)\log\left(1-\sigma(x_i^\top w)\right) \\
&= \sum_{i=1}^{n} y_i \log\left(\frac{1-\sigma(x_i^\top w)}{\sigma(x_i^\top w)}\right) - \log\left(1-\sigma(x_i^\top w)\right) \\
&= \sum_{i=1}^{n} -y_i x_i^\top w + \log\left(1+e^{x_i^\top w}\right)
\end{aligned}
$$

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

We obtain the following cost function we will minimize to learn the parameter w_*

$$\boxed{w_* = \arg\min L(w) := \sum_{i=1}^{n} -y_i x_i^\top w + \log\left(1+e^{x_i^\top w}\right)}$$

*If we are considering $y \in \{-1,1\}$, we will have a different function

** minimizing L is exactly equivalent to maximize the likelihood $\mathscr{L}$ since $p(X) \perp\!\!\!\perp w$

Gradient of the negative log likelihood

To minimize L, let's first look at its stationary points by computing its gradient:

$$\nabla L(w) = \nabla \left[\sum_{i=1}^{n} \log\left(1 + e^{x_i^\top w}\right) - y_i x_i^\top w \right] = \sum_{i=1}^{n} \frac{e^{x_i^\top w} x_i}{1 + e^{x_i^\top w}} - y_i x_i = \sum_{i=1}^{n} \left(\sigma(x_i^\top w) - y_i\right) x_i$$

Which can be written under the matrix form $\mathbf{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n \times d}$ and $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^{n}$

$$\nabla L(w) = \mathbf{X}^\top \left(\sigma(\mathbf{X}w) - \mathbf{y}\right)$$

- Same gradient as in LS but with σ
- No closed form solution to $\nabla L(w) = 0$
- Good news: the cost function L is convex

Convexity of the loss function L

Claim: The function

$$L(w) = \sum_{i=1}^{n} - y_i x_i^\top w + \log\left(1 + e^{x_i^\top w}\right)$$

is convex in the weight vector w

Proof: L is obtained through simple convexity preserving operations:

1. Positive combinations of convex function is convex
2. Composition of a convex and a linear functions is convex
3. A linear function is both convex and concave
4. $\eta \mapsto \log(1 + e^{\eta})$ is convex

Convexity of the loss function L

Claim: The function

$$L(w) = \sum_{i=1}^{n} - y_i x_i^\top w + \log\left(1 + e^{x_i^\top w}\right)$$

is convex in the weight vector w

Proof: L is obtained through simple convexity preserving operations:

1. Positive combinations of convex f
2. Composition of a convex and a lin
3. A linear function is both convex ar
4. $\eta \mapsto \log(1 + e^{\eta})$ is convex

Proof of 4: $h(\eta) := \log(1 + e^{\eta})$ is cvx

$$h'(\eta) = \frac{e^{\eta}}{1 + e^{\eta}} = \sigma(\eta)$$

$$h''(\eta) = \sigma'(\eta) = \frac{e^{\eta}}{(1 + e^{\eta})^2} \geq 0$$

Proof of the convexity of L

2. Composition of a convex and a linear functions is convex
4. $\eta \mapsto \log(1 + e^{\eta})$ is convex

$$\log\left(1 + e^{x_i^\top w}\right) \text{ is convex}$$

3. A linear function is both convex and concave

$$-y_i x_i^\top w \text{ is convex}$$

1. Positive combinations of convex function is convex

$$L(w) = \sum_{i=1}^{n} - y_i x_i^\top w + \log\left(1 + e^{x_i^\top w}\right) \text{ is convex}$$

Second proof: Hessian of L is psd

The Hessian $\nabla^2 L$ is the **matrix** whose entries are the **second derivatives** $\frac{\partial^2}{\partial w_i \partial w_j} L(w)$

$$\begin{aligned}
\nabla^2 L(w) &= \nabla [\nabla L(w)]^\top \\
&= \nabla \left[\textstyle\sum_{i=1}^n x_i \big(\sigma(x_i^\top w) - y_i \big) \right]^\top \\
&= \textstyle\sum_{i=1}^n \nabla \sigma(x_i^\top w) x_i^\top = \sum_{i=1}^n \sigma(x_i^\top w) \big(1 - \sigma(x_i^\top w) \big) x_i x_i^\top
\end{aligned}$$

It can be written under the matrix form:

$$\nabla^2 L(\theta) = \mathbf{X}^\top S \mathbf{X}, \quad \text{where } S = \text{diag}\left[\sigma(x_i^\top w) \big(1 - \sigma(x_i^\top w) \big) \right] \succcurlyeq 0$$

➡ L is convex since $\nabla^2 L(w) \succcurlyeq 0$

How to minimize the convex function L?

Gradient descent:

$$\begin{cases} w_0 \in \mathbb{R}^d \\ w_{t+1} = w_t - \gamma_t \nabla L(w_t) \end{cases}$$

can be slow

How to minimize the convex function L?

Gradient descent:

$$\begin{cases} w_0 \in \mathbb{R}^d \\ w_{t+1} = w_t - \gamma_t \sum_{i=1}^{n} \left(\sigma(x_i^\top) - y_i\right) x_i \end{cases}$$

can be slow

How to minimize the convex function L?

Gradient descent:

$$\begin{cases} w_0 \in \mathbb{R}^d \\ w_{t+1} = w_t - \gamma_t \sum_{i=1}^n \left(\sigma(x_i^\top) - y_i\right) x_i \end{cases}$$

can be slow

Stochastic gradient descent

$$\begin{cases} w_0 \in \mathbb{R}^d \\ w_{t+1} = w_t - \gamma_t n \left(\sigma(x_{i_t}^\top) - y_{i_t}\right) x_{i_t} \end{cases} \quad \text{where } \mathbb{P}[i_t = i] = 1/n$$

is faster but converges slower

Newton's method uses second order information

Newton's method **minimizes** the **quadratic approximation**:

$$L(w) \sim L(w_t) + \nabla L(w_t)^\top (w - w_t) + \frac{1}{2}(w - w_t)^\top \nabla^2 L(w_t)(w - w_t) := \phi_t(w)$$

$$w_{t+1} = \arg\min \phi_t(w) \implies \nabla L(w_t) + \nabla^2 L(w_t)(w_{t+1} - w_t) = 0$$

$$w_{t+1} = w_t - \gamma_t \nabla^2 L(w_t)^{-1} \nabla L(w_t)$$

The step-size is needed to ensure convergence (damped Newton's method)

The convergence is usually **faster than for gradient descent** but the **computational complexity is higher** (computing Hessian and solving a linear system)

Problem when the data are linearly separable

$$\inf_{w} L(w) = 0 = \lim_{\alpha \to \infty} L(\alpha \cdot \bar{w})$$

The inf value is not attained for a finite w

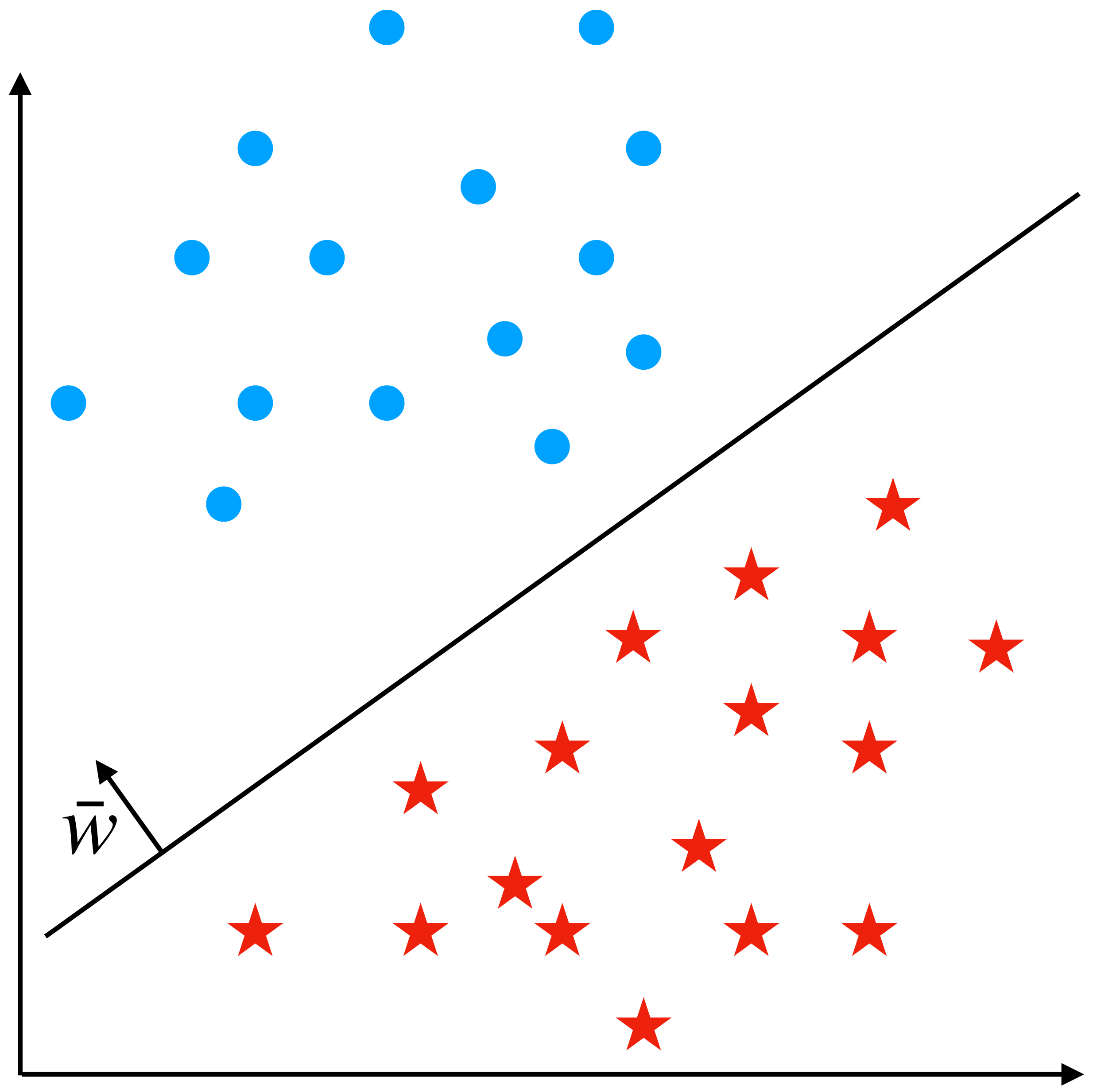
If we use an optimization algorithm, the weight will go to ∞

<u>Solution:</u> add a ℓ_2-regularization

➡ **ridge logistic regression:**

$$\sum_{i=1}^{n} -y_i x_i^\top w + \log\left(1 + e^{x_i^\top w}\right) + \lambda \|w\|_2^2$$

- Optimization perspective: stabilize the optimization process
- Statistical perspective: avoid overfitting



$$L(w) = \sum_{i=1}^{n} -y_i x_i^\top w + \log\left(1 + e^{x_i^\top w}\right)$$