4/22/20

Linear two-class Classification

P in/out pairs: $\{(\bar{x}_p, y_p)\}_{p=1}^{P}$ ← label

$\bar{x}_p \in R^N$;

$y_p \in \{0, +1\}$, OR $y_p \in \{-1, 1\}$

Two perspectives on classificaton

- regression perspective (extended view)
- perceptron " (view from "above")

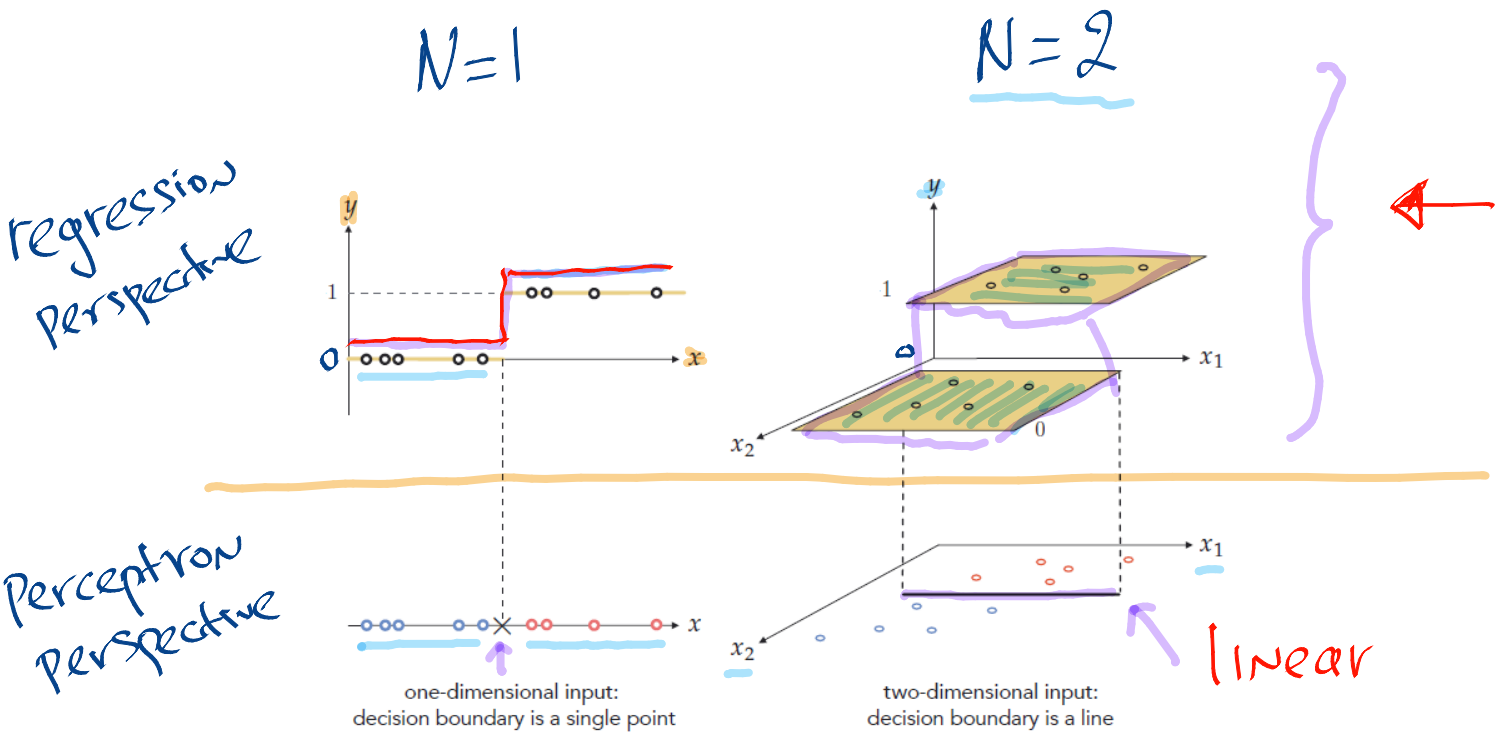
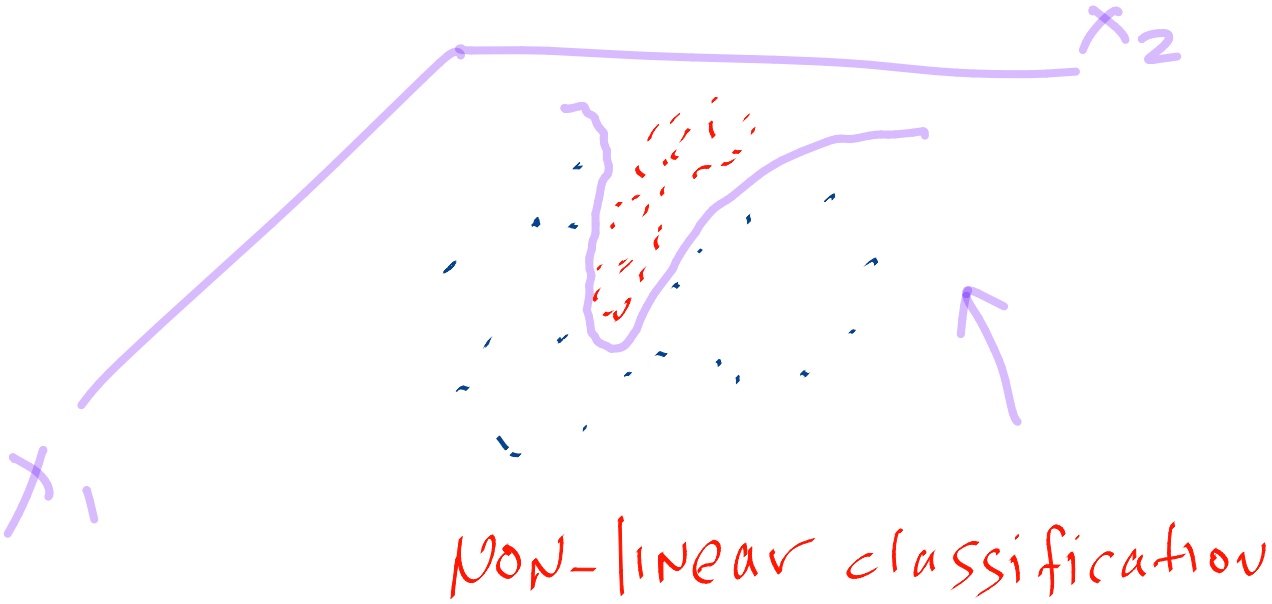


Figure 6.1 Two perspectives on classification illustrated using single-input (left column) and two-input (right column) toy datasets. The regression perspective shown in the top panels is equivalent to the perceptron perspective shown in the bottom panels, where we look at each respective dataset from "above." In the Perceptron perspective we also mark the decision boundary. This is where the step function (colored in yellow in the top panels) transitions from its bottom to top step. See text for further details.



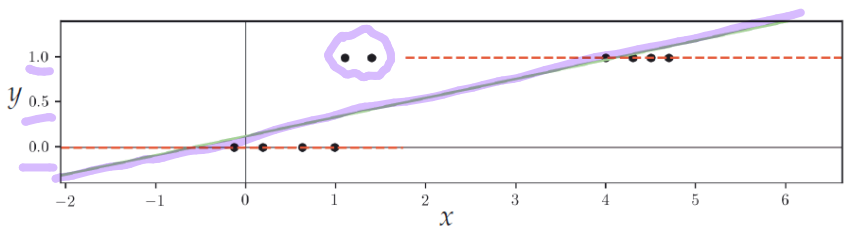
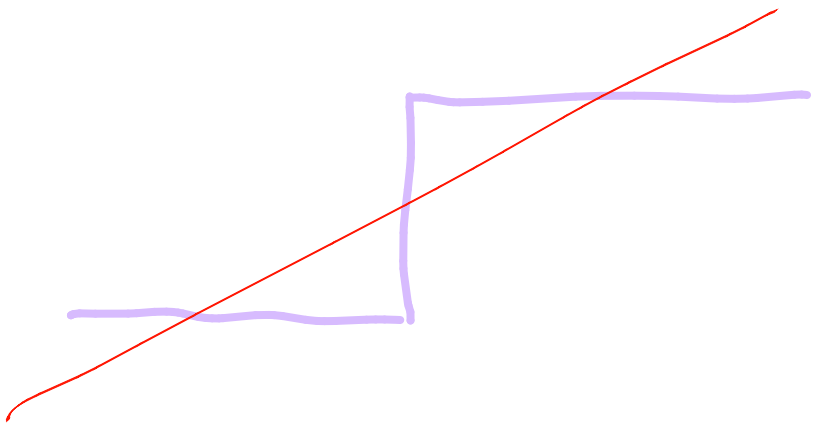


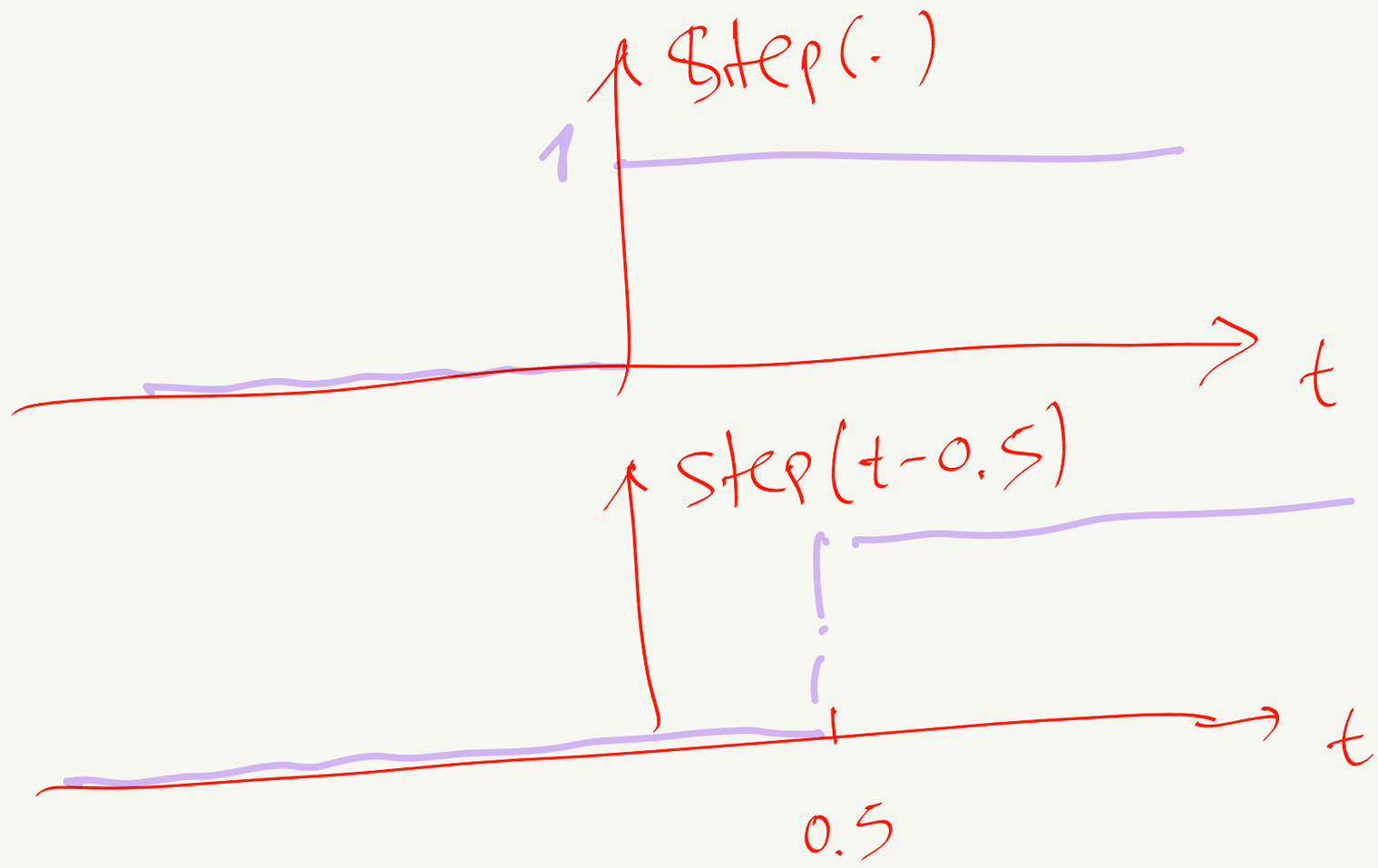
Figure 6.2 Figure associated with Example 6.1. See text for details.



pass hyperplane thru a step function

→ perform optimization

$$step(t) = \begin{cases} 1, & t > 0 \\ 0, & t < 0 \end{cases} \quad \text{(undefined for } t=0\text{)}$$

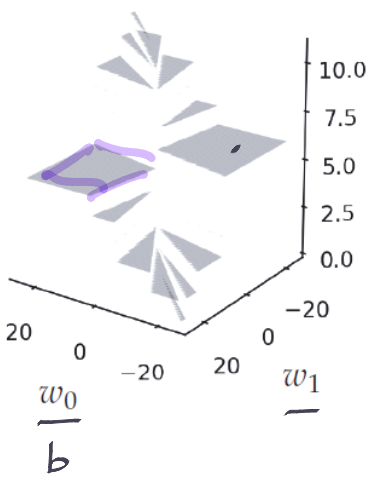


N-d

$$\tilde{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_N \end{bmatrix}_{(N+1)\times 1} \qquad \tilde{w} = \begin{bmatrix} b\,(w_0) \\ w_1 \\ \vdots \\ w_N \end{bmatrix}$$

(text: $\tilde{x} \to x$, $\tilde{w} \to w$)

hyperplane : $\bar{x}^T \tilde{w}$



b

$g(\tilde{w})$ for step(.)

$$\text{step}(\tilde{x}_p^T \tilde{w}) \approx y_p \;, \quad p = 1, \dots, P$$

One approach

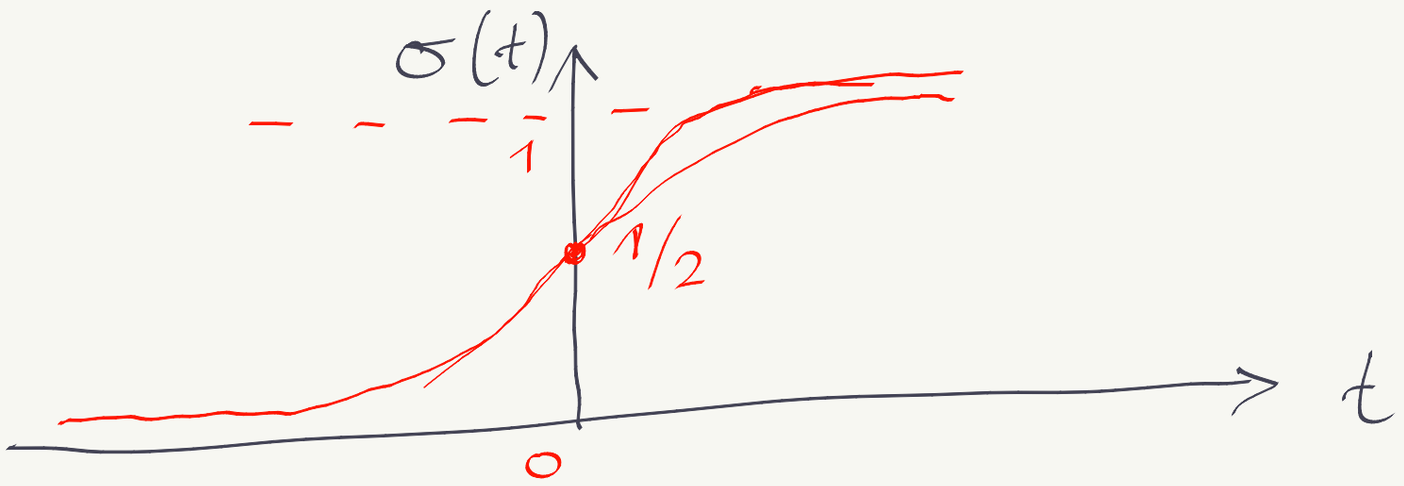
$$\Rightarrow \quad LS: \quad g(\tilde{w}) = \frac{1}{P} \sum_{p=1}^{P} \left(\text{step}(\tilde{x}_p^T \tilde{w}) - y_p \right)^2$$

Problematic

MAIN IDEA: approximate step function

Logistic sigmoid function.

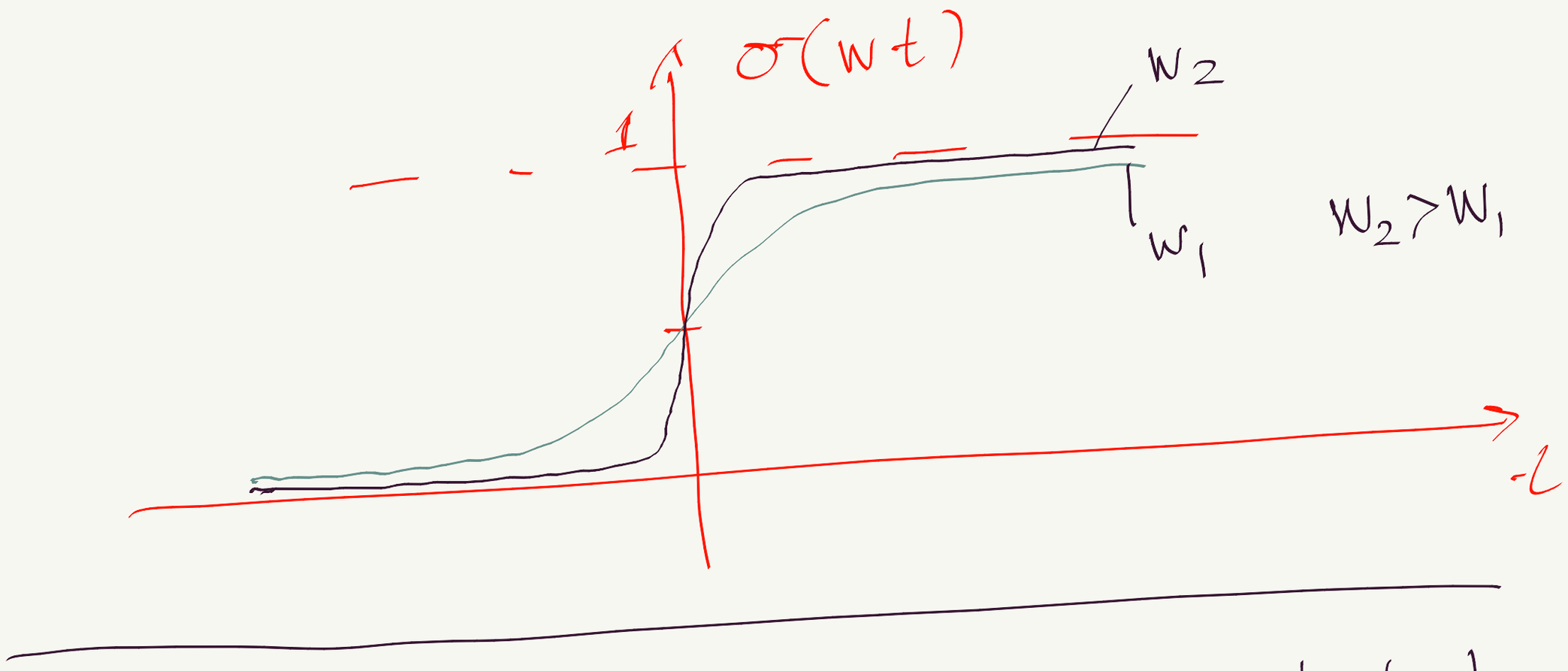
$$\sigma(t) = \frac{1}{1 + e^{-t}}$$



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

$$\lim_{t \to -\infty} \sigma(t) = \frac{1}{1 + e^{\infty}} = 0$$

$$t = 0 \qquad \sigma(0) = \frac{1}{1+1} = 1/2$$



19th century mathematician Verhulst

population growth in finite system

f: current population

1: max capacity of system

(1-f): remaining capacity

$$\rightarrow \frac{df}{dt} = f(1-f)$$

(we can easily show that σ(t) satisfies)

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

$$\sigma'(t) = \frac{-(-e^{-t})}{(1+e^{-t})^2} =$$

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

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

$$\boxed{\sigma'(t) = \sigma(t)\left[1-\sigma(t)\right]}$$

Logistic regression

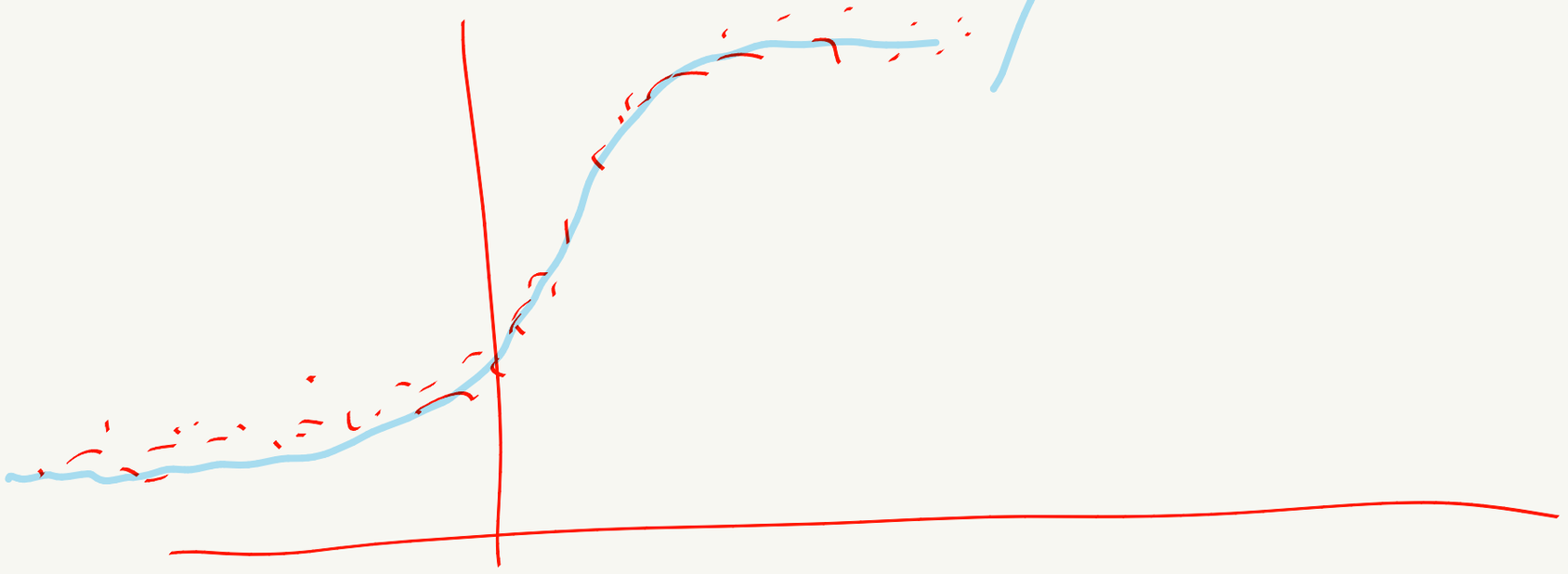
$$\sigma(\tilde{x}_p^T \tilde{w}) \simeq y_p \,, \quad p = 1, \dots, P$$

Cost function

A.

LS:

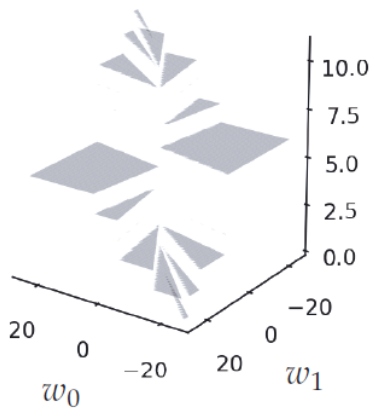
$$g(\tilde{w}) = \frac{1}{P} \sum_{p=1}^{P} \left(\sigma(\tilde{x}_p^T \tilde{w}) - y_p \right)^2$$

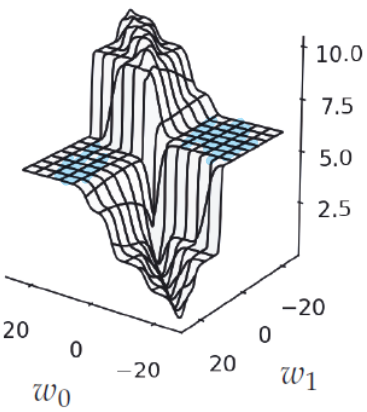


$g(\tilde{w})$:

- non-convex
- can be optimized using GD, Newton (w/ variations)

$g(\tilde{w})$ $g(\tilde{w})$





step $\sigma(t)$

Logistic regression

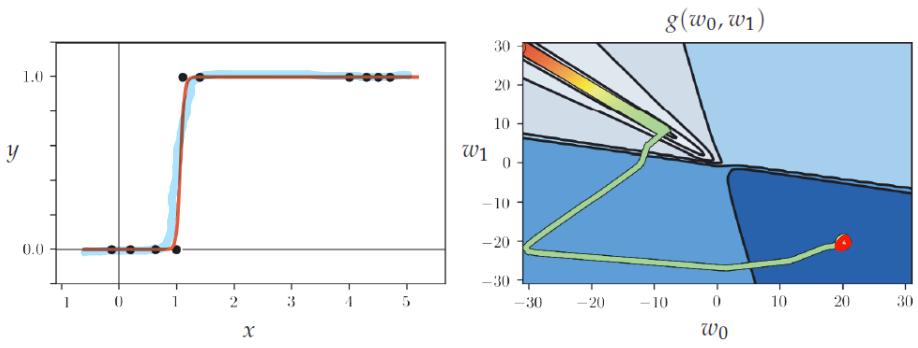


Figure 6.5 Figure associated with Example 6.3. See text for details.

GD

$w_0 = -w_1 = 20$

Normalized GD

Logistic Regression using Cross Entropy Cost

$y_p \in \{0, 1\}$

log error

$$g_p(\tilde{w}) = \begin{cases} -\log\left(\sigma\left(\tilde{x}_p^T \tilde{w}\right)\right), & y_p = 1 \\ -\log\left(1 - \sigma\left(\tilde{x}_p^T \tilde{w}\right)\right), & y_p = 0 \end{cases}$$

$g_p(\tilde{w})$ → (non-negetive, min = 0)

$$g_p(\tilde{w}) = -y_p \cdot \log \sigma\left(\tilde{x}_p^T \tilde{w}\right) - \underline{(1 - y_p)} \log\left(1 - \sigma\left(\tilde{x}_p^T \tilde{w}\right)\right)$$

$=$

$y_p = 1 \to 1$

$y_p = 0 \to 0$

0

1

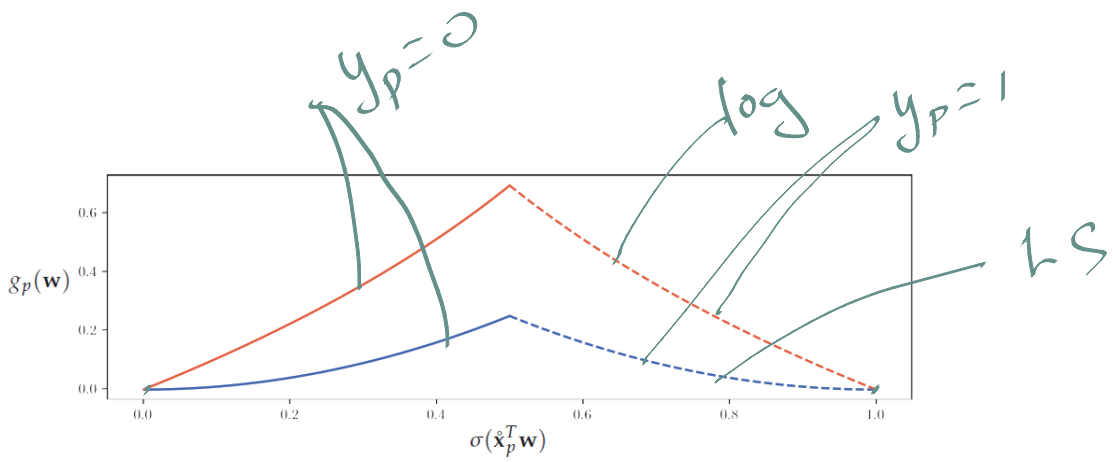


Figure 6.6 Visual comparison of the squared error (in blue) and the log error (in red) for two cases: $y_p = 0$ (solid curves) and $y_p = 1$ (dashed curves). In both cases the log error penalizes deviation from the true label value to a greater extent than the squared error.

Cross-entropy cost

$$g(\tilde{w}) = -\frac{1}{P}\sum_{p=1}^{P} y_p \log\left(\sigma(\tilde{x}_p^T\tilde{w})\right) + (1-y_p)\log\left(1-\sigma(\tilde{x}_p^T\tilde{w})\right)$$

logistic regression classifier using cross-entropy

$\Rightarrow$ CONVEX

$$\nabla_{\tilde{w}}\left(y_p \log\left(\sigma(\tilde{x}_p^T\tilde{w})\right)\right)$$

$$= y_p \underbrace{\frac{1}{\sigma(\tilde{x}_p^T\tilde{w})}}_{\text{scalar}} \cdot \sigma(\cdot)\left[1-\sigma(\cdot)\right] \cdot \nabla_{\tilde{w}}\left(\tilde{x}_p^T\tilde{w}\right)$$

$$\nabla_{\tilde{w}}\left(\tilde{x}_p^T\tilde{w}\right) = \tilde{x}_p$$

$$\nabla g(\tilde{w}) = -\frac{1}{P}\sum_{p=1}^{P}\left(y_p - \sigma(\tilde{x}^T\tilde{w})\right)\tilde{x}_p$$

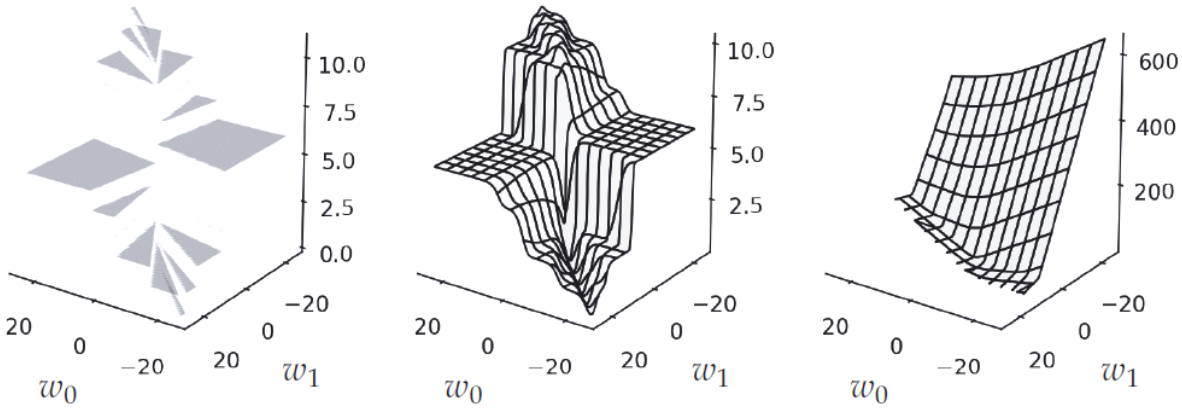


Figure 6.3 Figure associated with Example 6.2. See text for details.

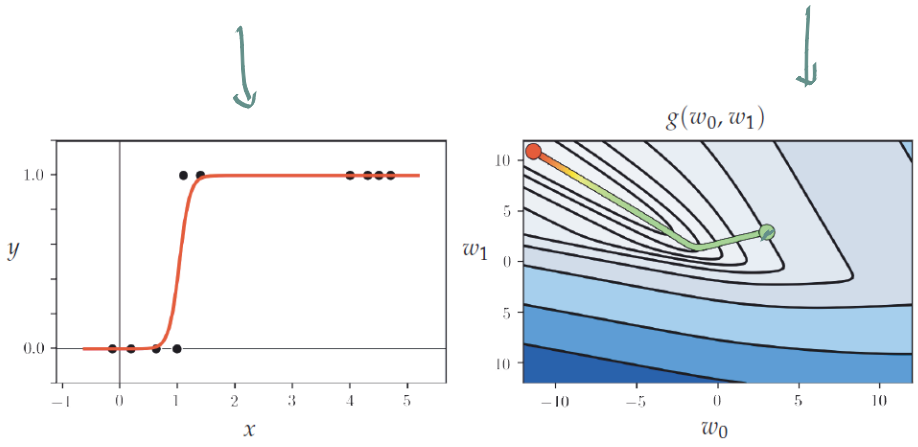


Figure 6.7 Figure associated with Example 6.4. See text for details.