Chapter ML:III

III. Linear Models

- Logistic Regression
- Loss Computation in Detail
- Overfitting
- Regularization
- Gradient Descent in Detail

Logistic Regression

Binary Classification Problems

Setting:

- X is a multiset of feature vectors from an inner product space $\mathbf{X}$, $\mathbf{X} \subseteq \mathbf{R}^p$.
- $C = \{0, 1\}$ is a set of two classes. Similarly: $\{-1, 1\}$, $\{\ominus, \oplus\}$, $\{\text{no}, \text{yes}\}$, etc.
- $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\} \subseteq X \times C$ is a multiset of examples.

Learning task:

- Fit D using a logistic function $y()$.

Logistic Regression

Binary Classification Problems

Setting:

- X is a multiset of feature vectors from an inner product space $\mathbf{X}$, $\mathbf{X} \subseteq \mathbf{R}^p$.
- $C = \{0, 1\}$ is a set of two classes. Similarly: $\{-1, 1\}$, $\{\ominus, \oplus\}$, $\{\text{no}, \text{yes}\}$, etc.
- $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\} \subseteq X \times C$ is a multiset of examples.

Learning task:

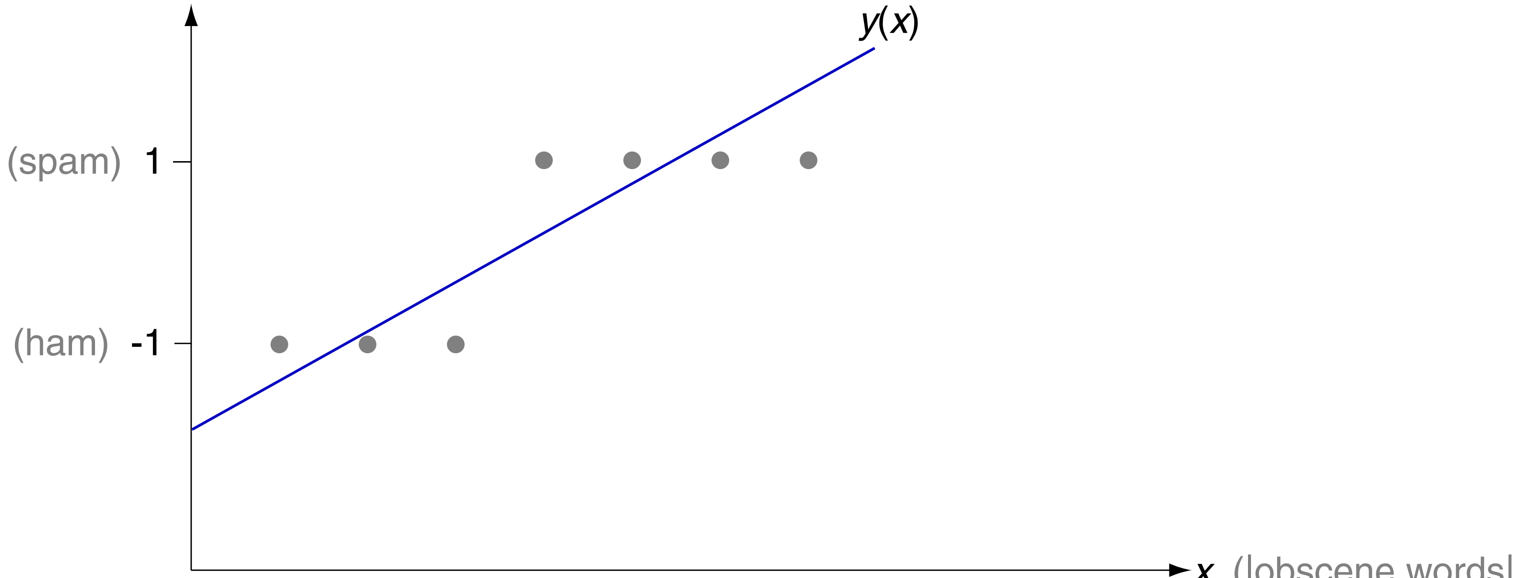
- Fit D using a logistic function $y()$.

Examples for binary classification problems:

- Is an email spam or ham?
- Is a patient infected or healthy?
- Is a bank customer creditworthy or not?

Logistic Regression

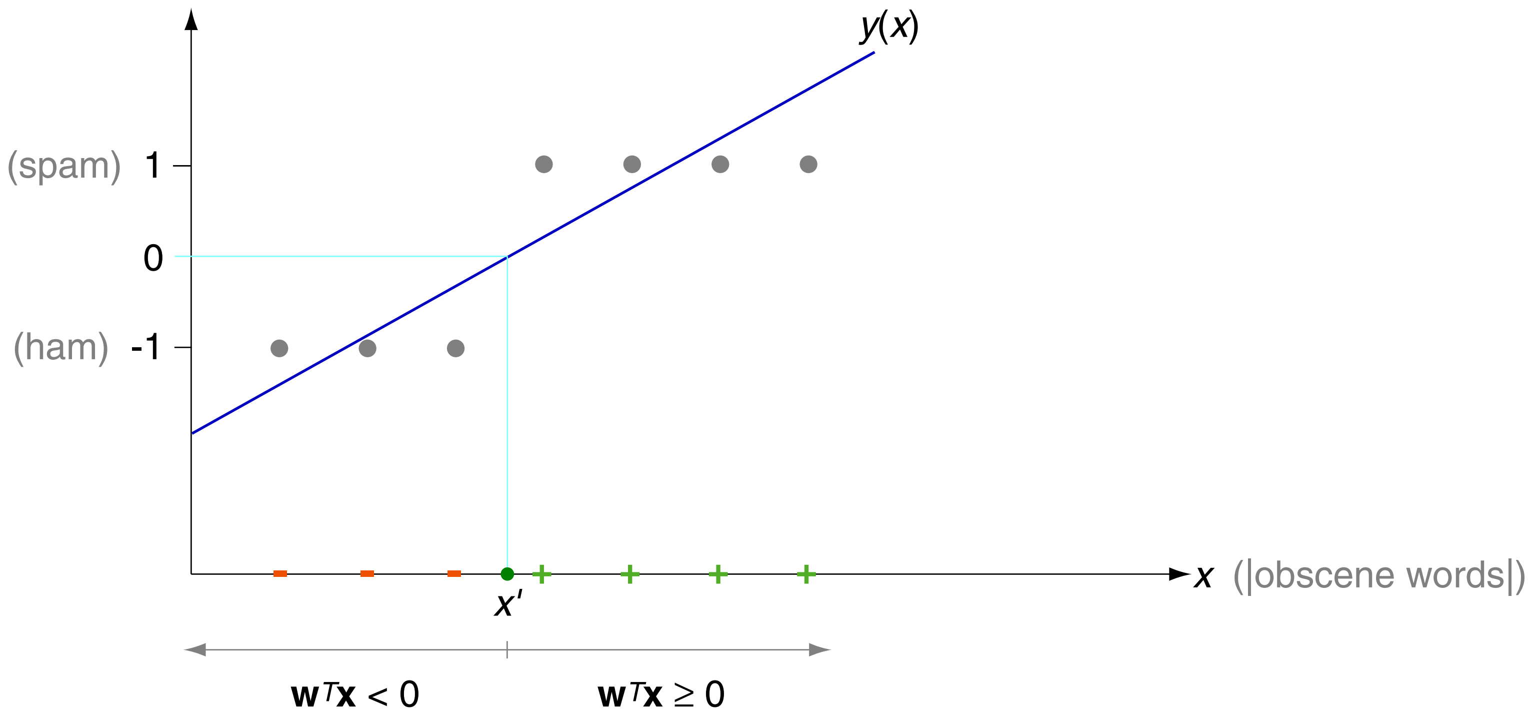
Linear Regression



- Linear regression: $y(\mathbf{x}) \overset{(\star)}{=} \mathbf{w}^T\mathbf{x}$

Logistic Regression

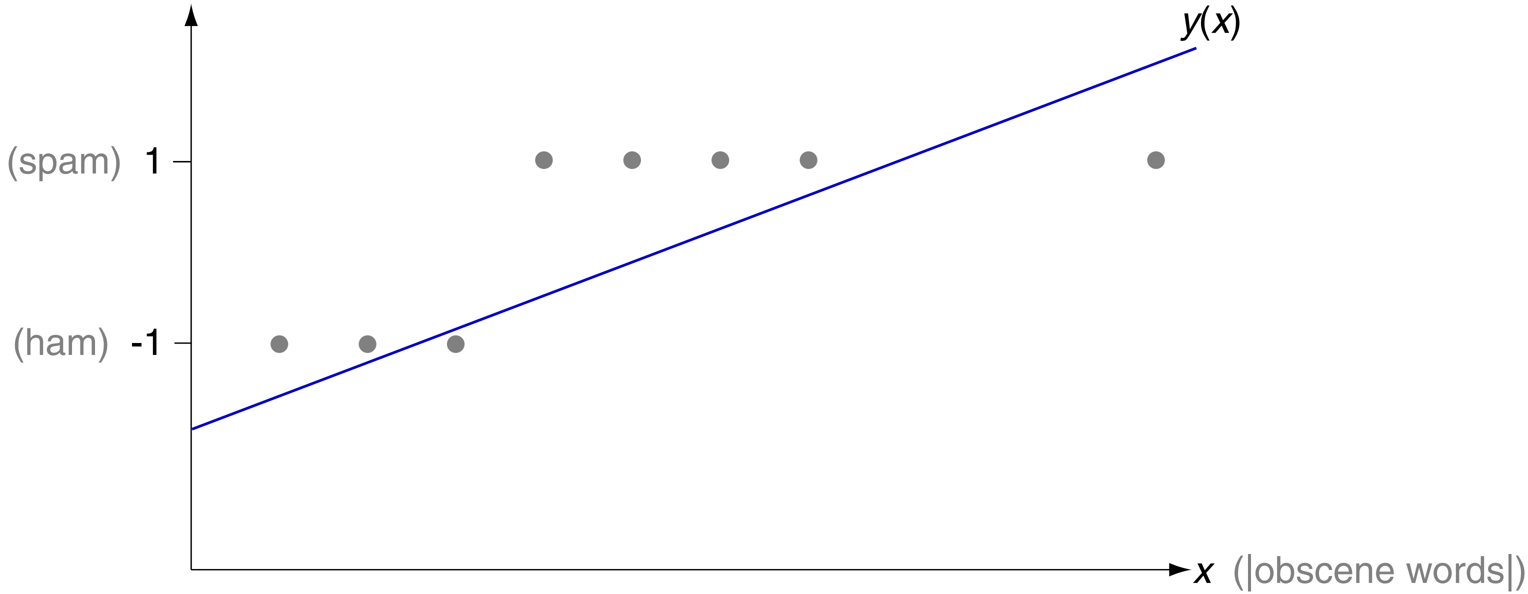
Linear Regression



- Linear regression: $y(\mathbf{x}) \overset{(\star)}{=} \mathbf{w}^T\mathbf{x}$
- Classification: Predict $\begin{cases} \text{“spam”}, & \text{if } \mathbf{w}^T\mathbf{x} \geq 0 \\ \text{“ham”}, & \text{if } \mathbf{w}^T\mathbf{x} < 0 \end{cases}$

Logistic Regression

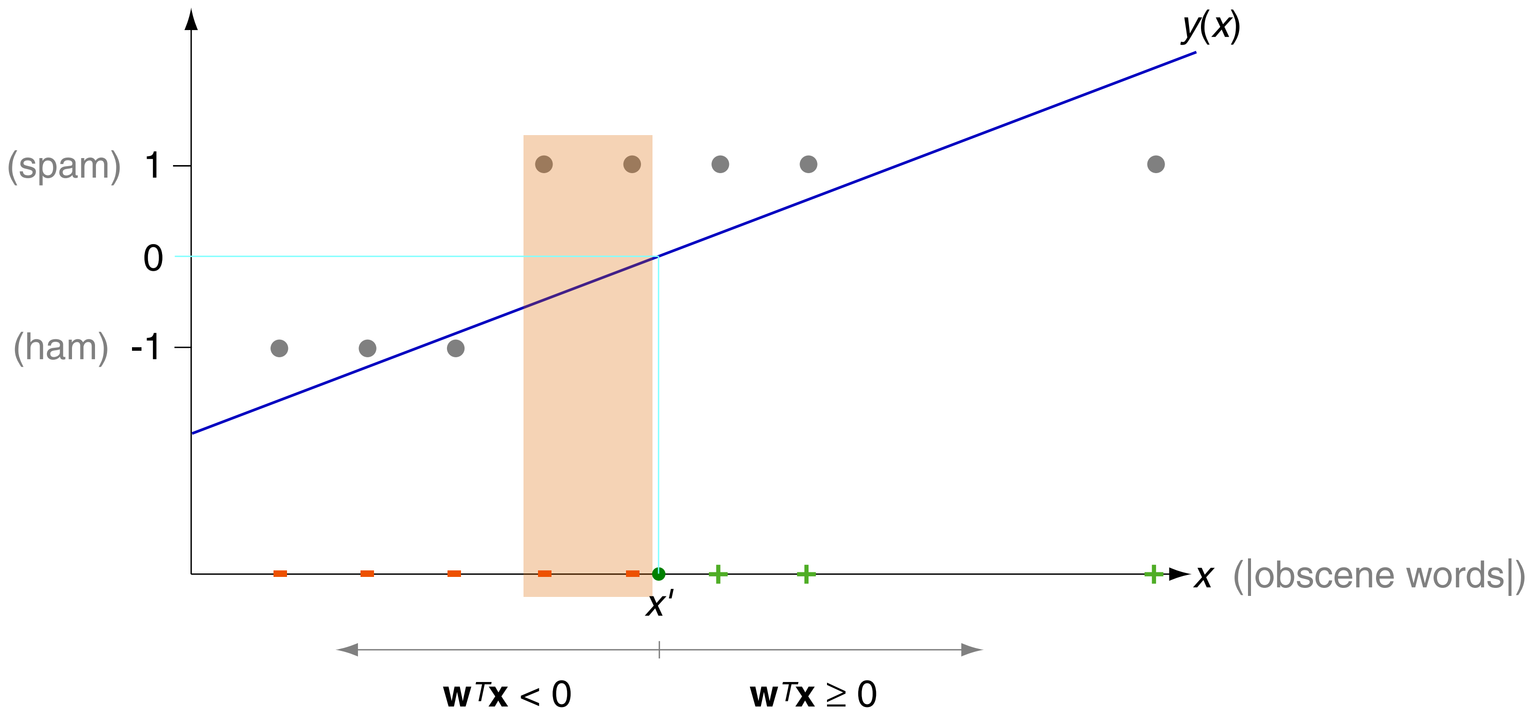
Linear Regression



- Linear regression: $y(\mathbf{x}) \stackrel{(\star)}{=} \mathbf{w}^T\mathbf{x}$
- Classification: Predict $\begin{cases} \text{“spam”}, & \text{if } \mathbf{w}^T\mathbf{x} \geq 0 \\ \text{“ham”}, & \text{if } \mathbf{w}^T\mathbf{x} < 0 \end{cases}$

Logistic Regression

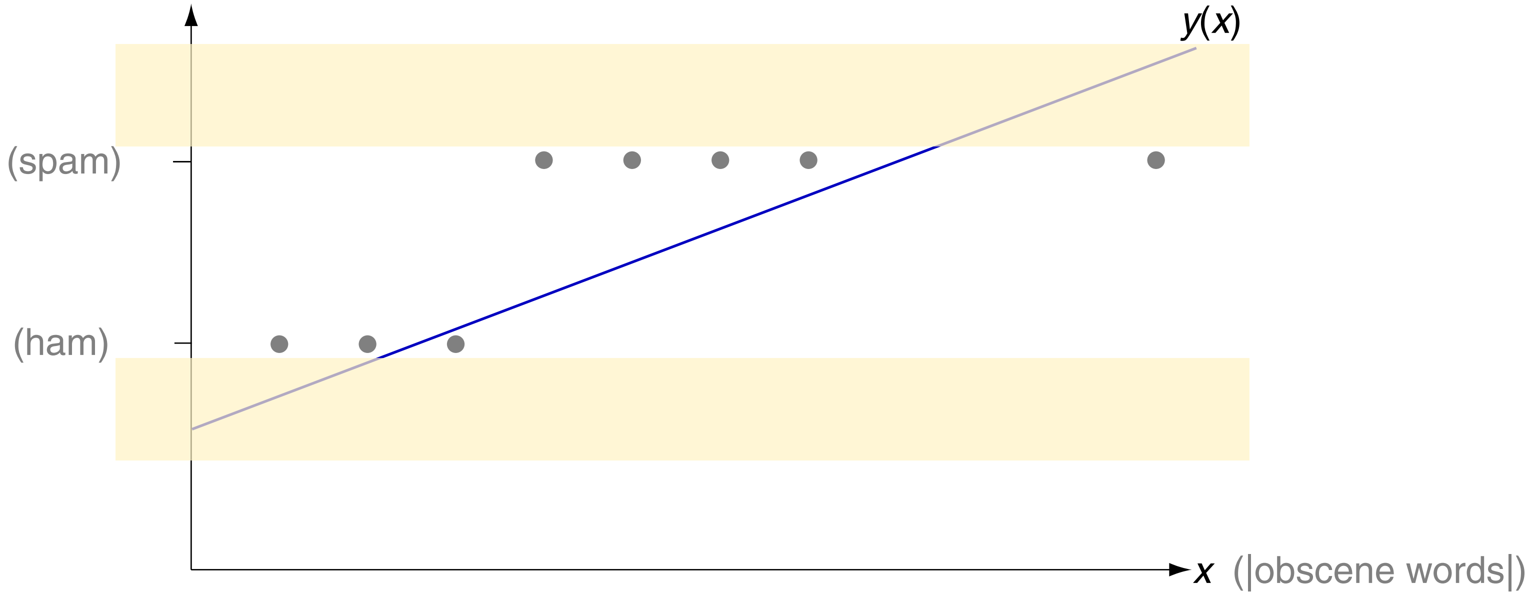
Linear Regression



- Linear regression: $y(\mathbf{x}) \overset{(\star)}{=} \mathbf{w}^T\mathbf{x}$
- Classification: Predict $\begin{cases} \text{“spam”}, & \text{if } \mathbf{w}^T\mathbf{x} \geq 0 \\ \text{“ham”}, & \text{if } \mathbf{w}^T\mathbf{x} < 0 \end{cases}$

Logistic Regression

Linear Regression



Restrict the range of $y(\mathbf{x})$ to reflect the two-class classification semantics:

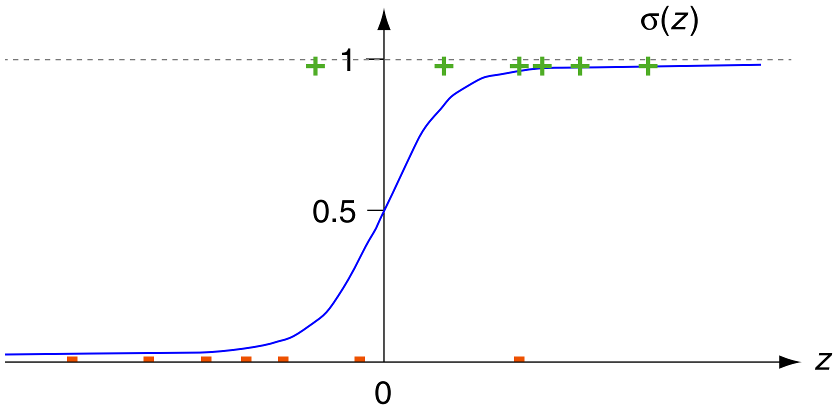
$$-1 \leq y(\mathbf{x}) \leq 1 \qquad \text{or} \qquad 0 \leq y(\mathbf{x}) \leq 1$$

Remarks:

- $(\star)$ Recap. We consider the feature vector $\mathbf{x}$ in its extended form when used as operand in a scalar product with the weight vector, $\mathbf{w}^T\mathbf{x}$, and consequently, when noted as argument of the model function, $y(\mathbf{x})$. I.e., $\mathbf{x} = (1, x_1, \ldots, x_p)^T \in \mathbf{R}^{p+1}$, and $x_0 = 1$.

Logistic Regression

Sigmoid (Logistic) Function

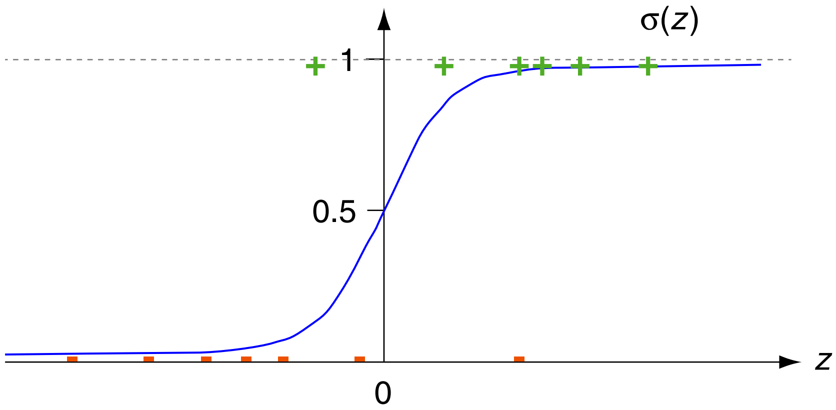


Sigmoid function

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

Logistic Regression

Sigmoid (Logistic) Function



Linear regression

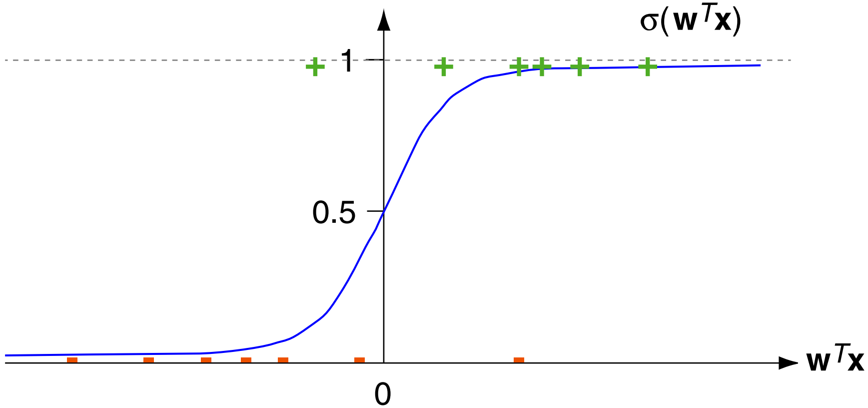
$$\mathbf{w}^T\mathbf{x}$$

Sigmoid function

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

Logistic Regression

Sigmoid (Logistic) Function

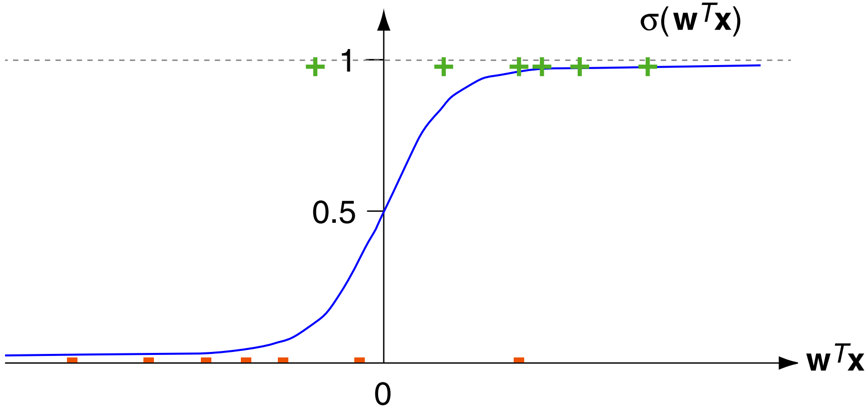


Linear regression | Sigmoid function | Logistic model function

$$\mathbf{w}^T\mathbf{x} \quad \circ \quad \sigma(z) = \frac{1}{1+e^{-z}} \quad \rightsquigarrow \quad y(\mathbf{x}) \equiv \sigma(\mathbf{w}^T\mathbf{x}) \overset{(\star)}{=} \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$$

Logistic Regression

Sigmoid (Logistic) Function



Linear regression		Sigmoid function		Logistic model function
$\mathbf{w}^T\mathbf{x}$	$\circ$	$\sigma(z) = \dfrac{1}{1+e^{-z}}$	$\rightsquigarrow$	$y(\mathbf{x}) \equiv \sigma(\mathbf{w}^T\mathbf{x}) \overset{(\star)}{=} \dfrac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$

$$y(\mathbf{x}) : \ \mathbf{R}^{p+1} \to (0;1)$$

Logistic Regression

Interpretation of the Logistic Model Function

$y(\mathbf{x}) \;=\; \sigma(\mathbf{w}^T\mathbf{x})$ is interpreted as the estimated probability for the event $C{=}1$:

- $y(\mathbf{x}) =: P(C{=}1 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(1 \mid \mathbf{x}; \mathbf{w})$, “Probability for $C{=}1$ given $\mathbf{x}$, parameterized by $\mathbf{w}$.”
- $1 - y(\mathbf{x}) =: P(C{=}0 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(0 \mid \mathbf{x}; \mathbf{w})$, “Probability for $C{=}0$ given $\mathbf{x}$, parameterized by $\mathbf{w}$.”

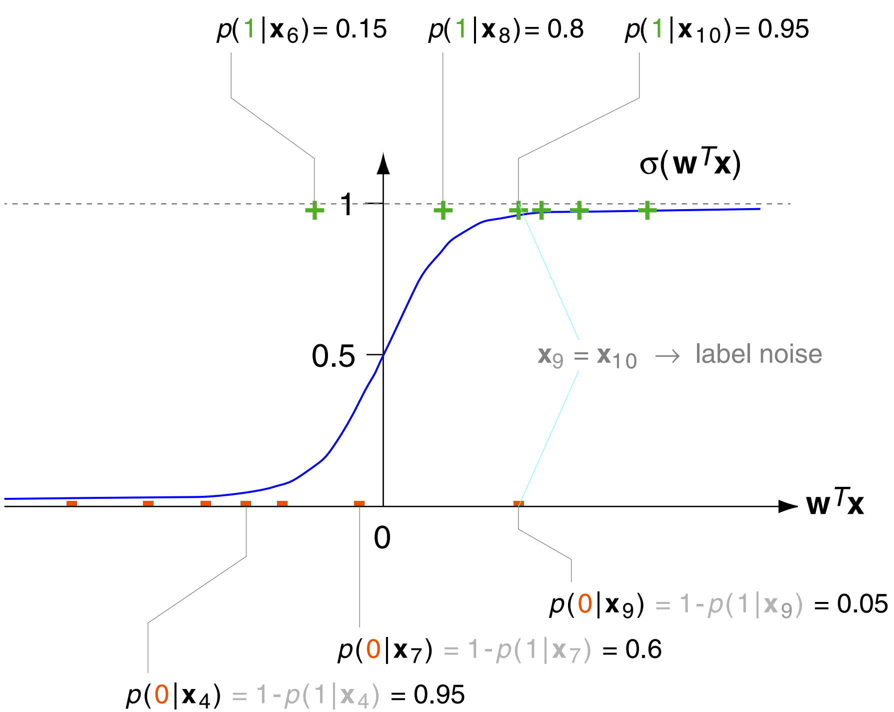
Logistic Regression

Interpretation of the Logistic Model Function

$y(\mathbf{x}) \;=\; \sigma(\mathbf{w}^T\mathbf{x})$ is interpreted as the estimated probability for the event $C{=}1$:

- $y(\mathbf{x}) =: P(C{=}1 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(1 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}1$ given x, parameterized by w."
- $1 - y(\mathbf{x}) =: P(C{=}0 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(0 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}0$ given x, parameterized by w."

The »+« and »-« are examples from D, projected on the z-axis, along with the probabilities as specified by $\sigma(\mathbf{w}^T\mathbf{x})$.



Logistic Regression

Interpretation of the Logistic Model Function (continued)

$y(\mathbf{x}) \;=\; \sigma(\mathbf{w}^T\mathbf{x})$ is interpreted as the estimated probability for the event $C{=}1$:

- $y(\mathbf{x}) =: P(C{=}1 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(1 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}1$ given $\mathbf{x}$, parameterized by $\mathbf{w}$."
- $1 - y(\mathbf{x}) =: P(C{=}0 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(0 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}0$ given $\mathbf{x}$, parameterized by $\mathbf{w}$."

Example (email spam classification) :

$$\mathbf{x} \;=\; \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \;=\; \begin{pmatrix} 1 \\ |\text{obscene words}| \end{pmatrix}, \qquad \mathbf{x}_1 = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \text{and} \quad y(\mathbf{x}_1) = 0.67$$

$\rightsquigarrow$ 67% chance that this email is spam.

Logistic Regression

Interpretation of the Logistic Model Function (continued)

$y(\mathbf{x}) \;=\; \sigma(\mathbf{w}^T\mathbf{x})$ is interpreted as the estimated probability for the event $C{=}1$:

- $y(\mathbf{x}) =: P(C{=}1 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(1 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}1$ given $\mathbf{x}$, parameterized by $\mathbf{w}$."
- $1 - y(\mathbf{x}) =: P(C{=}0 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(0 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}0$ given $\mathbf{x}$, parameterized by $\mathbf{w}$."

Example (email spam classification) :

$$\mathbf{x} \;=\; \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \;=\; \begin{pmatrix} 1 \\ |\text{obscene words}| \end{pmatrix}, \qquad \mathbf{x}_1 = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \text{and} \quad y(\mathbf{x}_1) = 0.67$$

$\leadsto$ 67% chance that this email is spam.

Logistic Regression

Interpretation of the Logistic Model Function (continued)

$y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$ is interpreted as the estimated probability for the event $C{=}1$:

- $y(\mathbf{x}) =: P(C{=}1 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(1 \mid \mathbf{x}; \mathbf{w})$,   “Probability for $C{=}1$ given $\mathbf{x}$, parameterized by $\mathbf{w}$.”
- $1 - y(\mathbf{x}) =: P(C{=}0 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(0 \mid \mathbf{x}; \mathbf{w})$,   “Probability for $C{=}0$ given $\mathbf{x}$, parameterized by $\mathbf{w}$.”

Example (email spam classification) :

$$\mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ |\text{obscene words}| \end{pmatrix}, \qquad \mathbf{x}_1 = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \text{and} \quad y(\mathbf{x}_1) = 0.67$$

$\rightsquigarrow$ 67% chance that this email is spam.

Classification: Predict $\begin{cases} 1, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \\ 0, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \end{cases}$

Logistic Regression

Interpretation of the Logistic Model Function (continued)

$y(\mathbf{x}) \;=\; \sigma(\mathbf{w}^T\mathbf{x})$ is interpreted as the estimated probability for the event $C{=}1$:

- $y(\mathbf{x}) =: P(C{=}1 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(1 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}1$ given x, parameterized by w."
- $1 - y(\mathbf{x}) =: P(C{=}0 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(0 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}0$ given x, parameterized by w."

Example (email spam classification) :

$$\mathbf{x} \;=\; \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \;=\; \begin{pmatrix} 1 \\ |\text{obscene words}| \end{pmatrix}, \qquad \mathbf{x}_1 = \begin{pmatrix} 1 \\ 5 \end{pmatrix} \quad \text{and} \quad y(\mathbf{x}_1) = 0.67$$

$\rightsquigarrow$ 67% chance that this email is spam.

Classification: Predict $\begin{cases} 1, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} \geq 0 \\ 0, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} < 0 \end{cases}$

Logistic Regression

Interpretation of the Logistic Model Function (continued)

$y(\mathbf{x}) \;=\; \sigma(\mathbf{w}^T\mathbf{x})$ is interpreted as the estimated probability for the event $C{=}1$:

- $y(\mathbf{x}) =: P(C{=}1 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(1 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}1$ given x, parameterized by w."
- $1 - y(\mathbf{x}) =: P(C{=}0 \mid \mathbf{X}{=}\mathbf{x}; \mathbf{w}) = p(0 \mid \mathbf{x}; \mathbf{w})$, "Probability for $C{=}0$ given x, parameterized by w."

Estimate optimum $\mathbf{w}$ by maximizing the probability $p(D; \mathbf{w})$:

$$\mathbf{w}_{\text{ML}} = \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}}\, p(D; \mathbf{w})$$

$\vdots$ [derivation]

$$= \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmin}}\, L_\sigma(\mathbf{w})$$

[RSS minimization]

I.e., optimizing $\mathbf{w}$ by maximizing $p(D; \mathbf{w})$ is equivalent to optimizing $\mathbf{w}$ by minimizing the logistic loss $L_\sigma(\mathbf{w})$.

Remarks (probabilistic view to classification) :

- If $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$ is interpreted as "probability for C=1 given feature vector $\mathbf{x}$", then $\mathbf{w}$ is the (unique) characterizing parameter vector of the hidden stochastic process that generates the observed data D.

  Recap. As a consequence, $\mathbf{w}$ is not the realization of a random variable—which would come along with a distribution—but an *exogenous parameter*, which is varied in order to find the maximum probability $p(D;\mathbf{w})$ or the minimum loss $L_\sigma(\mathbf{w})$.

  The fact that $\mathbf{w}$ is an exogenous parameter and not a the realization of a random variable is reflected by the notation, which uses a »;« instead of a »|« in $p()$.

- The underlying probability space—which can be left implicit—looks as follows:

  The sample space Ω corresponds to a set O of real-world objects, P is a probability measure defined on $\mathcal{P}(\Omega)$. The classification task (experiment) suggests two types of random variables, **X** $: \Omega \to \mathbf{X}$ and *C* $: \Omega \to \{0, 1\}$.

  See section Evaluating Effectiveness of part Machine Learning Basics for an illustration of the probabilistic view to classification, and section Probability Basics of part Bayesian Learning for a recap of concepts from probability theory.

- **X** and *C* denote (multivariate) random variables with ranges $\mathbf{X}$ and C respectively.

  X corresponds to a model formation function α that returns for a real-world object $o \in O$ its feature vector $\mathbf{x}$, $\mathbf{x} = \alpha(o)$, and *C* corresponds to an ideal classifier γ that returns its class c, $c = \gamma(o)$.

Remarks (probabilistic view to classification) : (continued)

- Interpreting $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$ as probability for the event $C{=}1$ means that $P(C{=}1 \mid \mathbf{X}{=}\mathbf{x})$ is defined as $y(\mathbf{x})$.

- The interpretation of $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$ as probability for the event $C{=}1$ is not a mathematical consequence but a decision of the modeler. This decision is based on the advantageous properties of the sigmoid function, on practical considerations, and on heuristic simplifications of the real world.

  Consider the following two aspects where an interpretation of $\sigma(\mathbf{w}^T\mathbf{x})$ is questionable:

  1. The sigmoid function implies that with $\mathbf{w}^T\mathbf{x} \to +\infty$ we get $y(\mathbf{x}) \to 1$ or $P(C{=}1) \to 1$. Likewise, with $\mathbf{w}^T\mathbf{x} \to -\infty$ we get $P(C{=}1) \to 0$. Though such a strict monotonicity appears self-evident, it need not necessarily correspond to the observed behavior in a real world experiment.

  2. The sigmoid function implies a smooth, virtually linear transition from low probability values (around 0.1) to high probability values (around 0.9) as its argument $\mathbf{w}^T\mathbf{x}$ increases.

     This link between the continuous growth of $\mathbf{w}^T\mathbf{x}$ and the continuous growth of probability values $P(C{=}1)$ presumes a proportional connection between cause (in the form of $\mathbf{X}{=}\mathbf{x}$) and effect (in the form of $C{=}1$). Again, such a relation appears sensible but may not necessarily model the real world.

Remarks (derivation of $L_\sigma(\mathbf{w})$) :

- The most probable (= optimum) hypothesis in the space H of possible hypotheses, h_{ML}, can be estimated with the maximum likelihood principle: $h_{\text{ML}} = \underset{h \in H}{\text{argmax}}\, p(D; h)$.

- Applied to logistic regression: $\mathbf{w}_{\text{ML}} = \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}}\, p(D; \mathbf{w})$, where

$$
\begin{aligned}
\underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}}\, p(D; \mathbf{w}) &= \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}} \prod_{(\mathbf{x},c) \in D} p(\mathbf{x}, c; \mathbf{w}) = \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}} \prod_{(\mathbf{x},c) \in D} \Big(p(c \mid \mathbf{x}; \mathbf{w}) \cdot p(\mathbf{x}) \Big) \\
&= \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}} \prod_{(\mathbf{x},c) \in D} p(\mathbf{x}) \cdot \prod_{(\mathbf{x},c) \in D} p(c \mid \mathbf{x}; \mathbf{w}) \overset{(1)}{=} \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}} \prod_{(\mathbf{x},c) \in D} p(c \mid \mathbf{x}; \mathbf{w}) \\
&= \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}} \prod_{(\mathbf{x},1) \in D} \sigma(\mathbf{w}^T\mathbf{x}) \cdot \prod_{(\mathbf{x},0) \in D} (1 - \sigma(\mathbf{w}^T\mathbf{x})) \\
&= \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}} \prod_{(\mathbf{x},1) \in D} y(\mathbf{x}) \cdot \prod_{(\mathbf{x},0) \in D} (1 - y(\mathbf{x})) \\
&\overset{(2)}{=} \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}} \log \prod_{(\mathbf{x},1) \in D} y(\mathbf{x}) + \log \prod_{(\mathbf{x},0) \in D} (1 - y(\mathbf{x})) \\
&= \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmax}} \sum_{(\mathbf{x},1) \in D} \log(y(\mathbf{x})) + \sum_{(\mathbf{x},0) \in D} \log(1 - y(\mathbf{x})) \\
&= \hookrightarrow \text{p. 24}
\end{aligned}
$$

Remarks (derivation of $L_\sigma(\mathbf{w})$) : (continued)

$$= \underset{\mathbf{w}\in\mathbf{R}^{p+1}}{\text{argmax}} \sum_{(\mathbf{x},c)\in D} \Big(c\cdot\log(y(\mathbf{x})) \;+\; (1-c)\cdot\log(1-y(\mathbf{x}))\Big)$$

$$\overset{(3)}{=} \underset{\mathbf{w}\in\mathbf{R}^{p+1}}{\text{argmin}} -\sum_{(\mathbf{x},c)\in D} \Big(c\cdot\log(y(\mathbf{x})) \;+\; (1-c)\cdot\log(1-y(\mathbf{x}))\Big)$$

$$\overset{(4)}{=} \underset{\mathbf{w}\in\mathbf{R}^{p+1}}{\text{argmin}} \sum_{(\mathbf{x},c)\in D} \overbrace{\Big(-c\cdot\log(y(\mathbf{x})) \;-\; (1-c)\cdot\log(1-y(\mathbf{x}))\Big)}^{l_\sigma\,:=}$$

$$=: \underset{\mathbf{w}\in\mathbf{R}^{p+1}}{\text{argmin}} \sum_{(\mathbf{x},c)\in D} l_\sigma(c, y(\mathbf{x})) \;=\; \underset{\mathbf{w}\in\mathbf{R}^{p+1}}{\text{argmin}}\; L_\sigma(\mathbf{w})$$

- Hints:

  (1) $\prod_{(\mathbf{x},c)\in D} p(\mathbf{x})$ is constant with respect to the variation of $\mathbf{w}$.

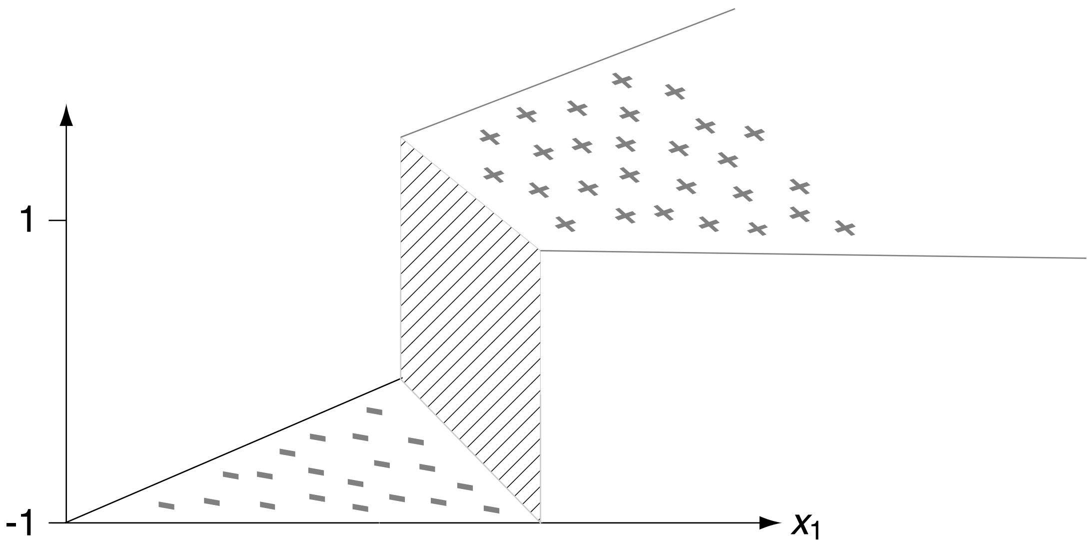
  (2) $\text{argmax}_x\, f(x) = \text{argmax}_x\, g\circ f(x)$ (similarly for argmin) if $g(z)$ is a strictly monotonically increasing function. Here, $\log(z)$ is in the role of $g(z)$. Conversely, if $g(z)$ is a strictly monotonically decreasing function, then $\text{argmax}_x\, f(x) = \text{argmin}_x\, g\circ f(x)$.

  (3) The maximization problem (the argmax-expression) can be reformulated as minimization problem, i.e., as an argmin-expression. See in this regard the second part of Hint (2).

  (4) The argument of the argmin-expression quantifies $L_\sigma(\mathbf{w})$, the global logistic loss related to some $\mathbf{w}$, and, analogously, the pointwise logistic loss, $l_\sigma(c, y(\mathbf{x}))$.

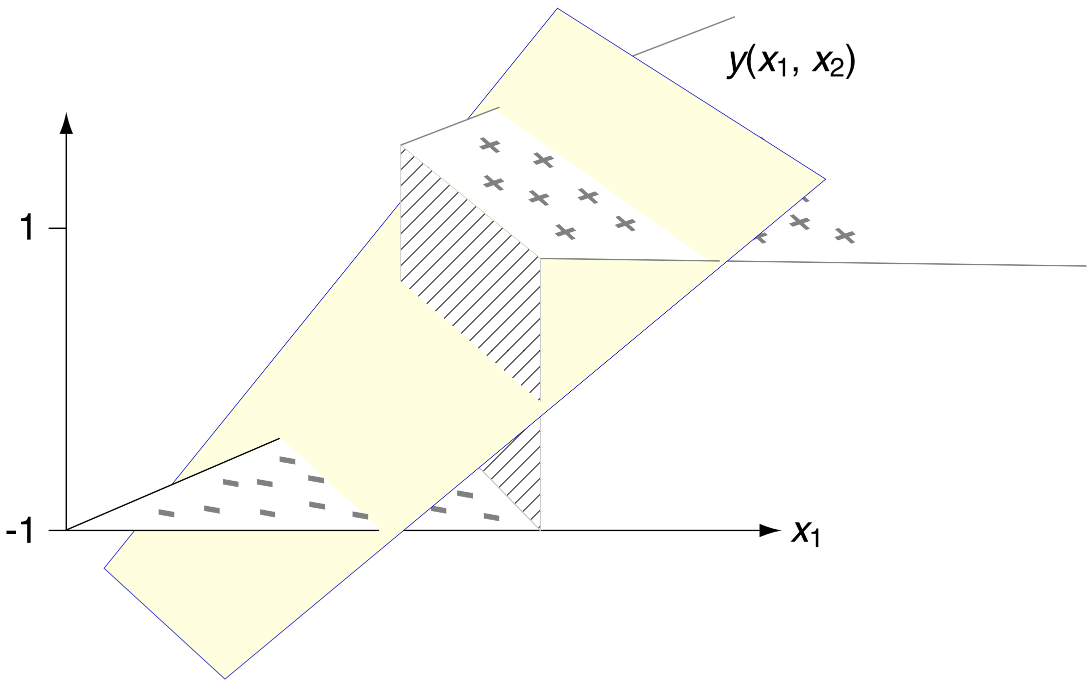
Logistic Regression

Recap. Linear Regression for Classification (illustrated for $p = 2$)



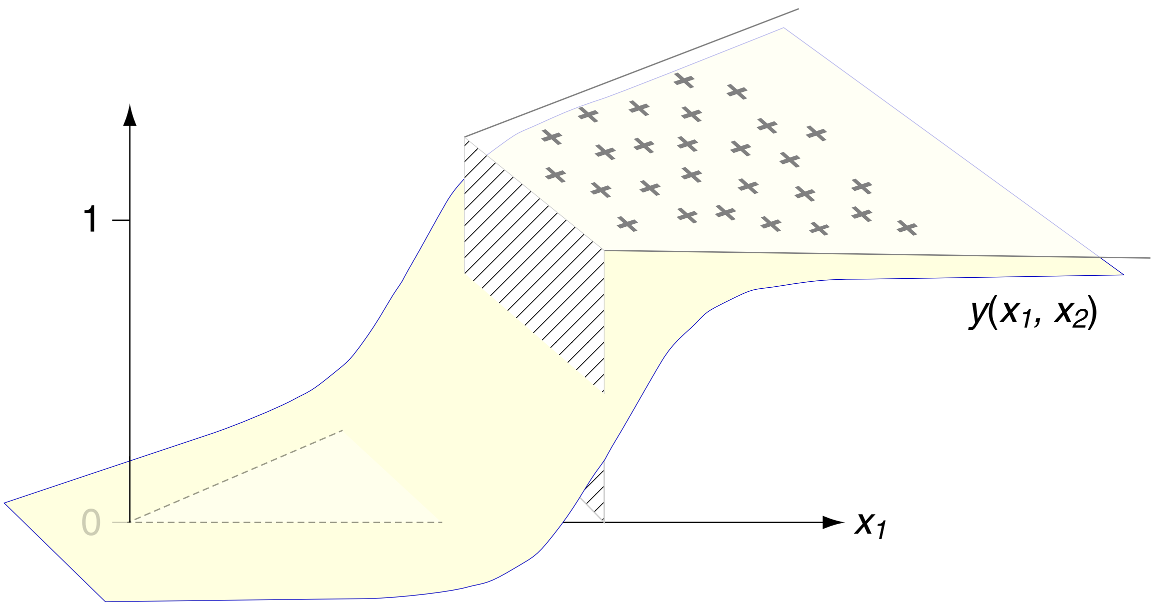
Logistic Regression

Recap. Linear Regression for Classification (illustrated for $p = 2$)



Logistic Regression

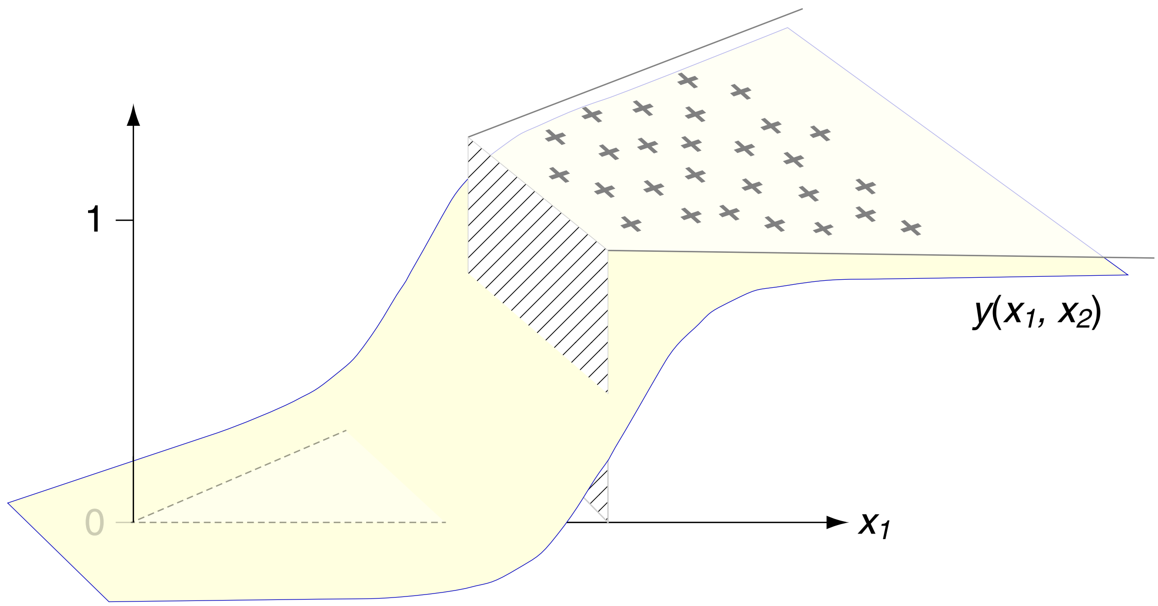
Logistic Regression for Classification (illustrated for $p = 2$)



Logistic Regression

Logistic Regression for Classification (illustrated for $p = 2$)

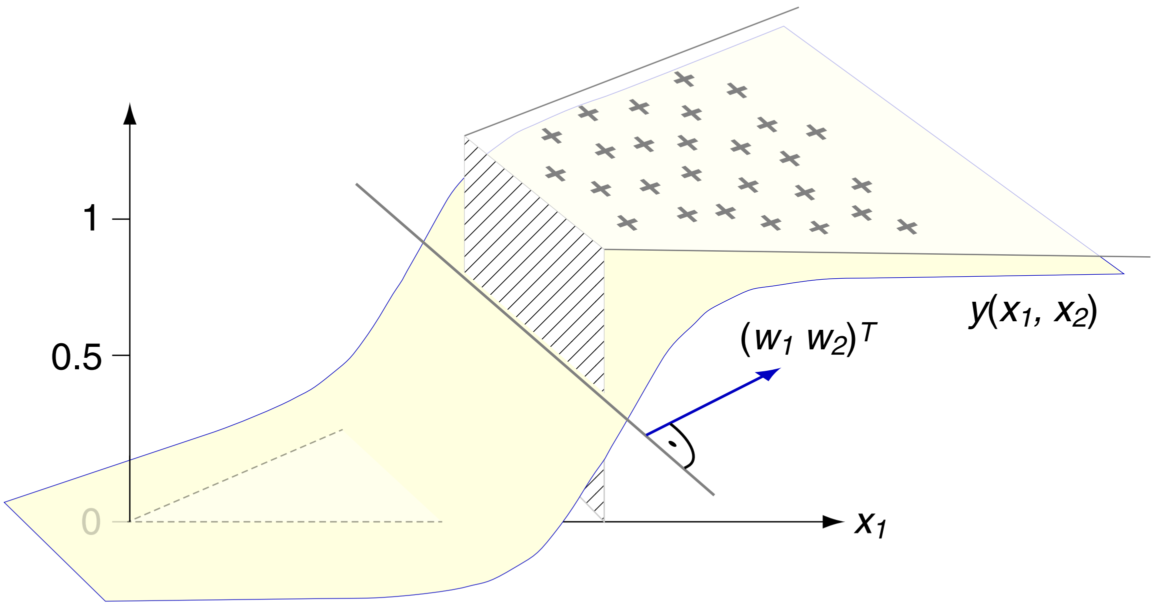
Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$.



$$y(x_1, x_2) = \frac{1}{1 + e^{-(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2)}}$$

Logistic Regression for Classification (illustrated for $p = 2$)

Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) \;=\; \dfrac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$.

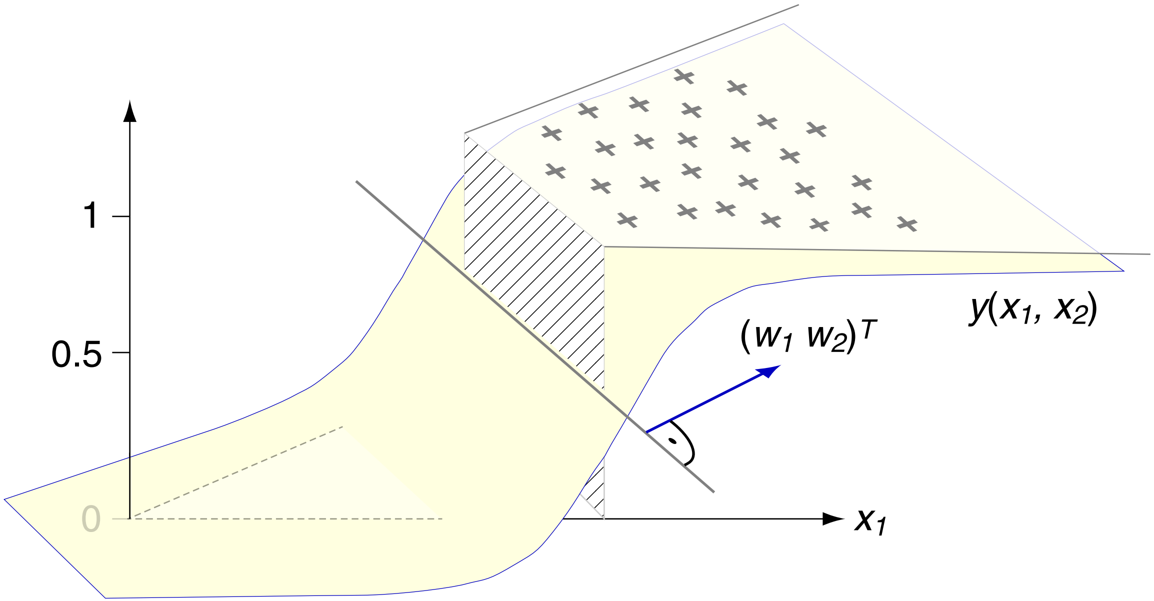


$$y(x_1, x_2) \;=\; \frac{1}{1+e^{-(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2)}}$$

Logistic Regression

Logistic Regression for Classification (illustrated for $p = 2$)

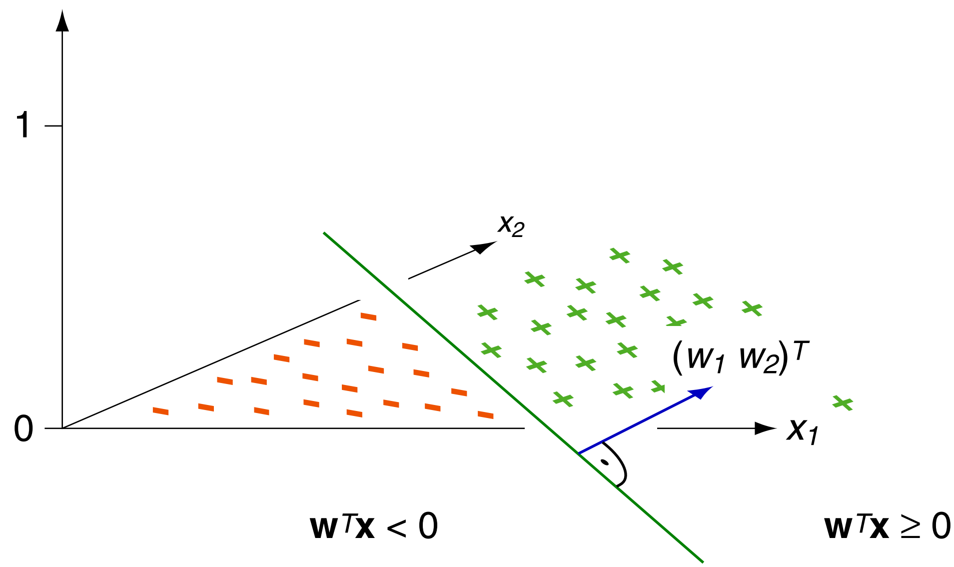
Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$.



Classification: Predict $\begin{cases} 1, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \\ 0, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \end{cases}$

Logistic Regression for Classification (illustrated for $p = 2$)

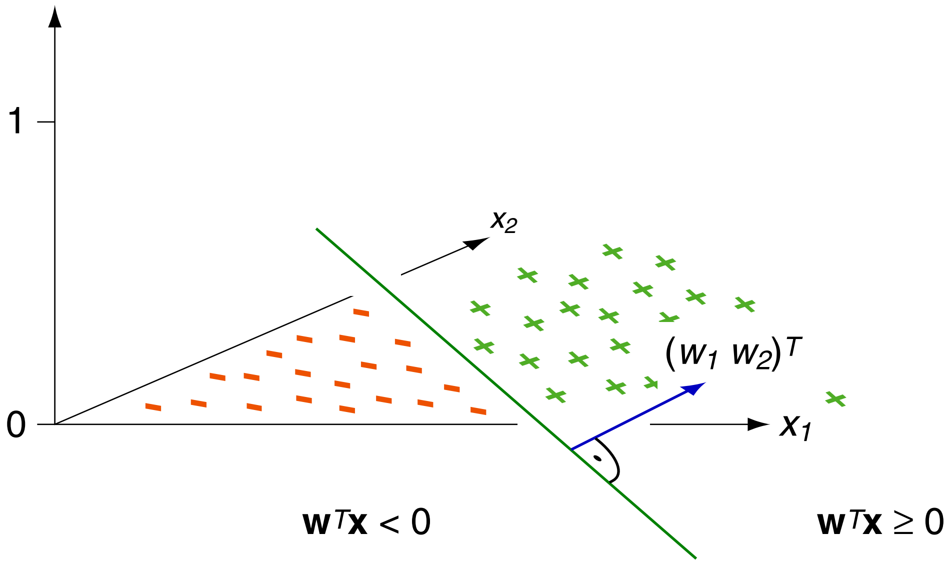
Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$.



Classification: Predict $\begin{cases} 1, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \\ 0, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \end{cases}$

Logistic Regression for Classification (illustrated for $p = 2$)

Use logistic regression to learn $\mathbf{w}$ from D, where $y(\mathbf{x}) = \dfrac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$.



Classification: Predict $\begin{cases} 1, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} \geq 0 \\ 0, & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) < 0.5 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} < 0 \end{cases}$

Logistic Regression

The BGD$_\sigma$ Algorithm [algorithms: LMS, BGD$_\sigma$, BGD, IGD, PT]

Algorithm:	BGD$_\sigma$	Batch Gradient Descent with logistic loss.
Input:	D	Multiset of examples $(\mathbf{x}, c)$ with $\mathbf{x} \in \mathbf{R}^p$, $c \in \{0, 1\}$.
	η	Learning rate, a small positive constant.
Output:	$\mathbf{w}$	Weight vector from $\mathbf{R}^{p+1}$. (= hypothesis)

BGD$_\sigma(D, \eta)$

```
initialize_random_weights(w),  t = 0
REPEAT
   t = t + 1,  Δw = 0
   FOREACH (x, c) ∈ D DO
      y(x) =(⋆) σ(w^T x) = 1 / (1 + e^(−w^T x))
      δ = c − y(x)
      Δw =(⋆) Δw + η · δ · x      // −δ · x is the derivative of l_σ(c, y(x)) wrt. w.
   ENDDO
   w = w + Δw
UNTIL(convergence(D, y(), t))
return(w)
```

Logistic Regression

The BGD$_\sigma$ Algorithm [algorithms: LMS, BGD$_\sigma$, BGD, IGD, PT]

Algorithm:	BGD$_\sigma$	Batch Gradient Descent with logistic loss.
Input:	D	Multiset of examples $(\mathbf{x}, c)$ with $\mathbf{x} \in \mathbf{R}^p$, $c \in \{0, 1\}$.
	η	Learning rate, a small positive constant.
Output:	$\mathbf{w}$	Weight vector from $\mathbf{R}^{p+1}$. (= hypothesis)

BGD$_\sigma(D, \eta)$

initialize_random_weights$(\mathbf{w})$, $t = 0$
REPEAT
  $t = t + 1$, $\Delta\mathbf{w} = 0$
  FOREACH $(\mathbf{x}, c) \in D$ DO
    Model function evaluation.
    Calculation of residual.
    Calculation of derivative of the loss, accumulate for D.
  ENDDO
  Parameter vector update $\hat{=}$ one gradient step down.
UNTIL(*convergence*$(D, y(), t)$)
return$(\mathbf{w})$

Remarks (BGD$_\sigma$ Algorithm) :

- The BGD$_\sigma$ Algorithm is an iterative method to estimate $\mathbf{w}_{\text{ML}}$ in the model function $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$. There is no direct method (such as the normal equations in linear regression) to tackle the optimization problem.

- The BGD$_\sigma$ Algorithm exploits the derivative of the pointwise logistic loss $l_\sigma(c, y(\mathbf{x}))$ with respect to $\mathbf{w}$, which is $-\delta \cdot \mathbf{x} = -(c - y(\mathbf{x})) \cdot \mathbf{x} = -(c - \sigma(\mathbf{w}^T\mathbf{x})) \cdot \mathbf{x}$. The derivation of this term, as well as notes regarding the speed of convergence of the basic gradient descent are given in section Gradient Descent in Detail of part Linear Models.

- Each BGD$_\sigma$ iteration "`REPEAT ... UNTIL`"
  1. computes the direction of steepest loss descent as

     $$-\nabla L_\sigma(\mathbf{w}_t) = \sum_{(\mathbf{x},c)\in D} (c - y_t(\mathbf{x})) \cdot \mathbf{x}, \text{ and}$$

  2. updates $\mathbf{w}_t$ by taking a step of length η in this direction.

- The output of the logistic model (function), $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x})$, is a continuous variable in the domain $(0, 1)$, while the dependent variable that we use for training is Bernoulli-distributed: $c = 0 \ \vee \ c = 1$.

Remarks: (continued)

- Recap. A hypothesis is a proposed explanation for a phenomenon. [Wikipedia]

  Here, hypothesis "explains" (= fits) the data D. Hence, a concrete model function $y()$, $\mathbf{y}()$, or, if the function type is clear from the context, its parameters $\mathbf{w}$ or $\boldsymbol{\theta}$ are called "hypothesis". The variable name h (similarly: h_1, h_2, h_i, h', etc.) may be used to refer to a specific instance of a model function or its parameters.

- Recap. The function *convergence*$()$ can analyze the global logistic loss, $L_\sigma(\mathbf{w}_t)$, or the norm of the loss gradient, $||\nabla L_\sigma(\mathbf{w}_t)||$, and compare it to a small positive bound ε. Consider in this regard the vectors of observed and computed classes, $D|_c$ and $y(D|_\mathbf{x})$ respectively. In addition, the function may check via t an upper bound on the number of iterations.

Logistic Regression

ML Stack: Logistic Regression

[ML stack: LMS, log. regression, loss comp., regularization, GD]

Optimization approach

Optimization objective

Loss function [+ Regularization]

Model function $\rightsquigarrow$ Hypothesis space

Task

Data

ML Stack: Logistic Regression

[ML stack: LMS, log. regression, loss comp., regularization, GD]

Optimization approach

Optimization objective

Loss function [+ Regularization]

Model function $\rightsquigarrow$ Hypothesis space

Task

Data

Binary classification

$$D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\} \subseteq X \times \{0, 1\}$$

ML Stack: Logistic Regression

[ML stack: LMS, log. regression, loss comp., regularization, GD]

Optimization approach

Optimization objective

Loss function [+ Regularization]

Model function $\rightsquigarrow$ Hypothesis space

- Hypothesis space: $\mathbf{w} \in \mathbf{R}^{p+1}$
- Logistic model: $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$

Task

Binary classification

Data

$D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\} \subseteq X \times \{0, 1\}$

ML Stack: Logistic Regression

[ML stack: LMS, log. regression, loss comp., regularization, GD]

Optimization approach

Optimization objective

Loss function [+ Regularization]

Model function $\rightsquigarrow$ Hypothesis space

Task

Data

- Objective: minimize logistic loss $L_\sigma(\mathbf{w})$
- Regularization: none
- Loss: $l_\sigma(c, y(\mathbf{x})) = -c \cdot \log(y(\mathbf{x})) - (1-c) \cdot \log(1-y(\mathbf{x}))$

- Hypothesis space: $\mathbf{w} \in \mathbf{R}^{p+1}$
- Logistic model: $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$

Binary classification

$D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\} \subseteq X \times \{0, 1\}$

Logistic Regression

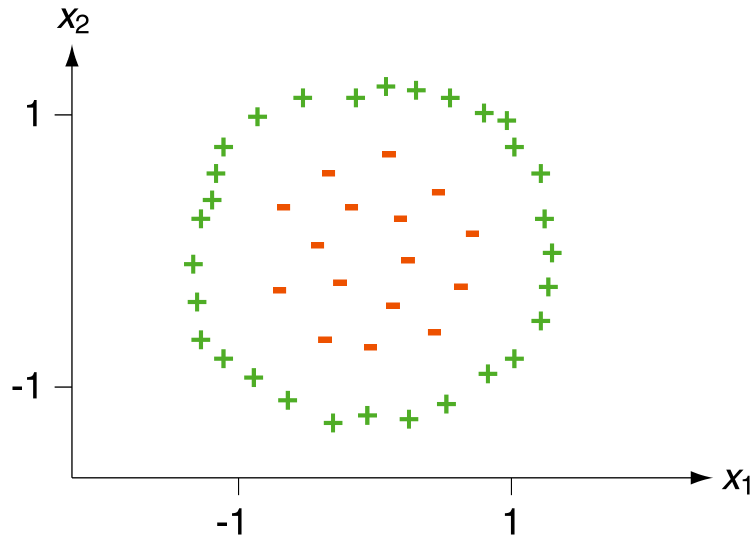
ML Stack: Logistic Regression

[ML stack: LMS, log. regression, loss comp., regularization, GD]

Optimization approach	BGD, Newton-Raphson, BFGS, Conjugate GD
Optimization objective Loss function [+ Regularization]	❑ Objective: minimize logistic loss $L_\sigma(\mathbf{w})$ ❑ Regularization: none ❑ Loss: $l_\sigma(c, y(\mathbf{x})) = -c \cdot \log(y(\mathbf{x})) - (1-c) \cdot \log(1-y(\mathbf{x}))$
Model function ⇝ Hypothesis space	❑ Hypothesis space: $\mathbf{w} \in \mathbf{R}^{p+1}$ ❑ Logistic model: $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$
Task	Binary classification
Data	$D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\} \subseteq X \times \{0, 1\}$

Logistic Regression

Non-Linear Decision Boundaries [linear regression]

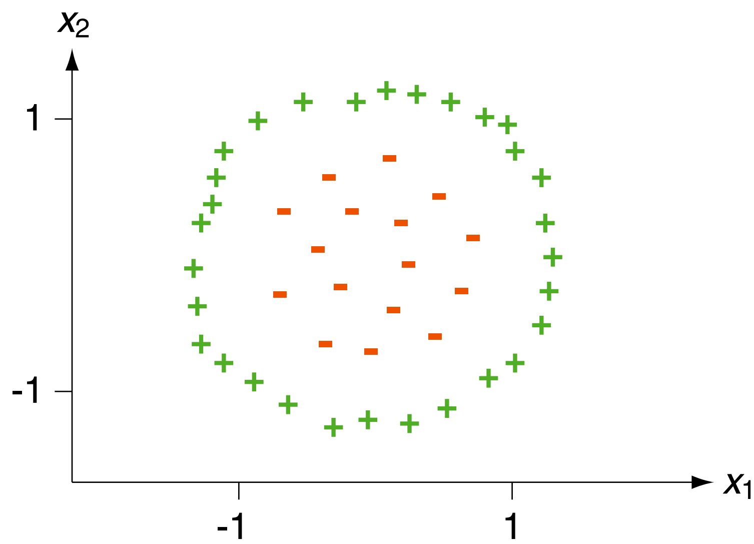


Higher order polynomial terms in the features (linear in the parameters):

$$y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_1^2 + w_4 \cdot x_2^2)$$

Logistic Regression

Non-Linear Decision Boundaries [linear regression]

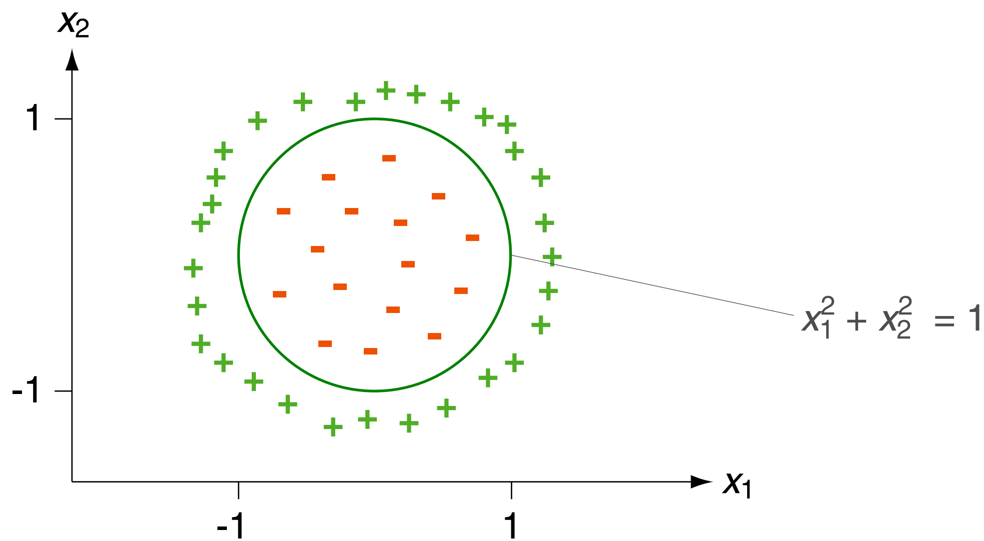


Higher order polynomial terms in the features (linear in the parameters):

$$y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_1^2 + w_4 \cdot x_2^2)$$

Logistic Regression

Non-Linear Decision Boundaries [linear regression]



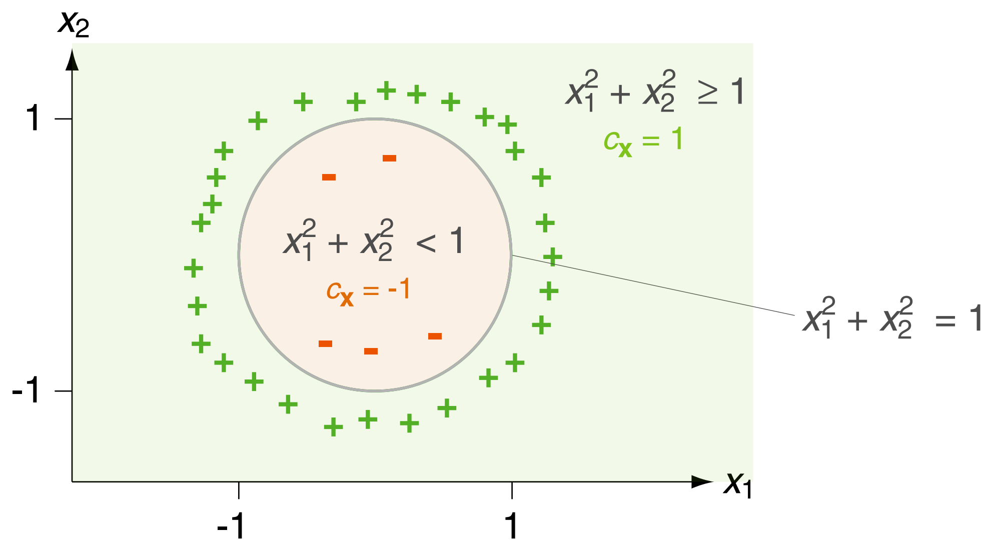
Higher order polynomial terms in the features (linear in the parameters):

$$y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_1^2 + w_4 \cdot x_2^2)$$

with $\mathbf{w} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad \rightsquigarrow \quad y(\mathbf{x}) = \dfrac{1}{1 + e^{-(-1 \,+\, x_1^2 \,+\, x_2^2)}}$

Logistic Regression

Non-Linear Decision Boundaries [linear regression]



Higher order polynomial terms in the features (linear in the parameters):

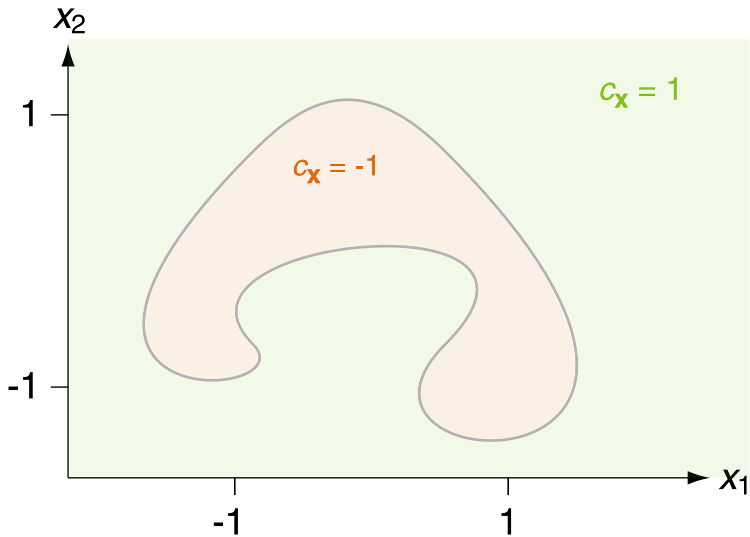
$$y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_1^2 + w_4 \cdot x_2^2)$$

with $\mathbf{w} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$ $\quad\rightsquigarrow\quad y(\mathbf{x}) = \dfrac{1}{1 + e^{-(-1 \, + \, x_1^2 \, + \, x_2^2)}}$

Classification: Predict $\begin{cases} 1, & \text{if } x_1^2 + x_2^2 \geq 1 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} \geq 0 \\ 0, & \text{if } x_1^2 + x_2^2 < 1 \quad \Leftrightarrow \quad \mathbf{w}^T\mathbf{x} < 0 \end{cases}$

Logistic Regression

Non-Linear Decision Boundaries [linear regression]



More complex polynomials entail more complex decision boundaries:

$$y(\mathbf{x}) = \sigma(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_1^2 + w_4 \cdot x_1^2 \cdot x_2 + w_5 \cdot x_1^2 \cdot x_2^2 + \ldots)$$

Remarks:

- Under logistic regression the structure of a hypothesis, i.e., the forms of possible decision boundaries, is identical to the hypothesis structure under linear regression. Similarly, the respective hypothesis spaces are the same.

  Hence, the expressiveness, i.e., the complexity of classification problems that can be tackled (or, the effectiveness at which classification problems can be decided) is identical for the two regression approaches.

- Linear regression and logistic regression differ in the way how the model function parameters, $\mathbf{w}$, are determined. In both cases the optimum $\mathbf{w}$ is the result of a loss minimization problem.

  Recall that "loss" means "interpretation of residuals." Linear regression and logistic regression differ with respect to this interpretation: While the former simply squares the residuals, this way putting a high weight onto outliers, the latter models an increasing confidence in class membership probability with increasing hyperplane distance. This different interpretation will usually lead to a different parameter vector $\mathbf{w}$, i.e., a different hyperplane.

Chapter ML:III

III. Linear Models

Loss Computation in Detail

ML Stack: Loss Computation

[ML stack: LMS, log. regression, loss comp., regularization, GD]

Optimization approach

⋮

Optimization objective

Loss function [+ Regularization]

- Objective: minimize loss
- Regularization: none
- Loss: 0/1 loss,
  squared loss,
  logistic loss,
  hinge loss

Model function ⇝ Hypothesis space

Task

⋮

Data

⋮

Remarks (loss function) :

- Given a hypothesis $\mathbf{w}$, its (global) loss, $L(\mathbf{w})$, tells us something about the effectiveness of $\mathbf{w}$. When used as sole criterion (e.g., no regularization is applied) we select from two hypotheses that with the smaller loss. I.e., the most effective hypothesis is found by loss minimization.

  Conversely, we call a function, whose minimization determines the most effective hypothesis, a loss function.

- Loss functions can be distinguished with respect to the problem class they are typically applied to: regression versus classification. Keep in mind that this distinction is not unique since loss functions with continuous output are applied to classification problems as well.

- Furthermore, we distinguish the

  1. *pointwise loss* $l(c, y(\mathbf{x}))$, which is computed for a single $\mathbf{x}$, and the
  2. *global loss* $L(\mathbf{w})$, which accumulates the pointwise losses of all $\mathbf{x} \in X$ for the weight vector $\mathbf{w}$ used in a specific $y()$:

  $$L(\mathbf{w}) = \sum_{(\mathbf{x},c) \in D} l(c, y(\mathbf{x}))$$

  The pointwise loss is also called per-example loss. [p.268, Goodfellow/Bengio/Courville 2016]

- Instead of “loss” (function, computation) also the terms “error” (function, computation), “cost” (function, computation), or “performance” (function, computation) are used, usually with the same semantics as introduced here. We will use the term “error” for classification problems and the term “loss” for both classification and regression problems.

Remarks (different roles of loss functions) :

- Observe that loss functions are employed at two places (in two roles) in an optimization approach:
  1. For the fitting of the data (i.e., the parameter update during regression / optimization / hyperplane search), where a new position of the hyperplane is computed.

     Example: Lines 6+7 in the BGD$_\sigma$ Algorithm, which is illustrated here over the hyperplane distance (red curve in the figure at the bottom right).
  2. For the evaluation of the effectiveness of a hypothesis, where the proportion of correctly and misclassified examples is analyzed.

     Example: Line 10 in the BGD$_\sigma$ Algorithm.
     General: section Evaluating Effectiveness of part Machine Learning Basics.

  Typically, (1) fitting (optimization) and (2) effectiveness evaluation are done with different loss functions. E.g., logistic regression uses L_σ and $L_{0/1}$ for fitting and evaluation respectively. However, linear regression (not classification) uses RSS (the L_2 loss) for both fitting and evaluation. The basic perceptron learning algorithm uses the misclassification information (the $L_{0/1}$ loss) for both fitting and evaluation.

Loss Computation in Detail

(1) Linear Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:

  – 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) = I_{\neq}(c, \text{sign}(y(\mathbf{x}))) = \begin{cases} 0 & \text{if } c = \text{sign}(y(\mathbf{x})) \\ 1 & \text{otherwise} \end{cases}$

  – Squared loss. $l_2(c, y(\mathbf{x})) = (c - y(\mathbf{x}))^2$

Loss Computation in Detail

(1) Linear Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:

  - 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) \;=\; I_{\neq}(c, \mathrm{sign}(y(\mathbf{x}))) \;=\; \begin{cases} 0 & \text{if } c = \mathrm{sign}(y(\mathbf{x})) \\ 1 & \text{otherwise} \end{cases}$
  - Squared loss. $l_2(c, y(\mathbf{x})) \;=\; (c - y(\mathbf{x}))^2$

Loss Computation in Detail

(1) Linear Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

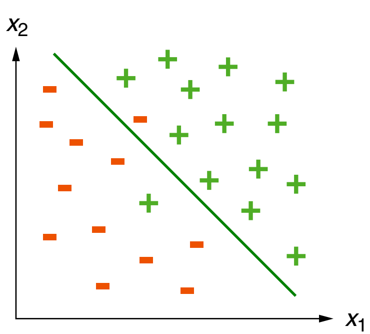
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:

  – 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) = I_{\neq}(c, \text{sign}(y(\mathbf{x}))) = \begin{cases} 0 & \text{if } c = \text{sign}(y(\mathbf{x})) \\ 1 & \text{otherwise} \end{cases}$

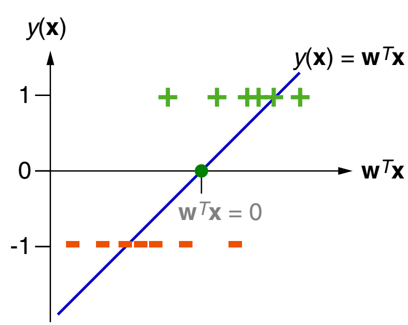
  – Squared loss. $l_2(c, y(\mathbf{x})) = (c - y(\mathbf{x}))^2$

Illustration for a particular $\mathbf{w}$:

Input space:



$y()$ over hyperplane distance:



Loss Computation in Detail

(1) Linear Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

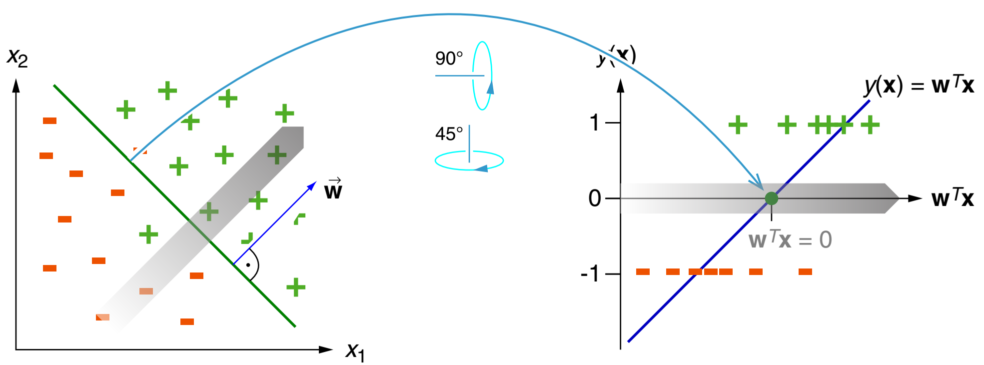
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:

  - 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) = I_{\neq}(c, \text{sign}(y(\mathbf{x}))) = \begin{cases} 0 & \text{if } c = \text{sign}(y(\mathbf{x})) \\ 1 & \text{otherwise} \end{cases}$

  - Squared loss. $l_2(c, y(\mathbf{x})) = (c - y(\mathbf{x}))^2$

Illustration for a particular $\mathbf{w}$:

Input space: $y()$ over hyperplane distance:



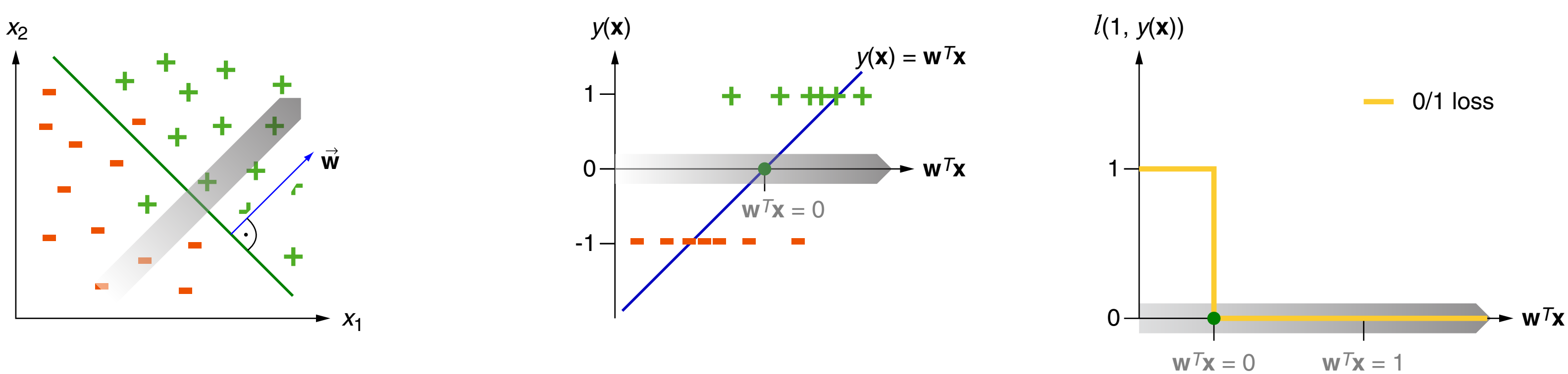
Loss Computation in Detail

(1) Linear Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:
  - 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) = I_{\neq}(c, \text{sign}(y(\mathbf{x}))) = \begin{cases} 0 & \text{if } c = \text{sign}(y(\mathbf{x})) \\ 1 & \text{otherwise} \end{cases}$
  - Squared loss. $l_2(c, y(\mathbf{x})) = (c - y(\mathbf{x}))^2$

Illustration for a particular $\mathbf{w}$:

Input space: $y()$ over hyperplane distance: Loss over hyperplane distance:



Loss Computation in Detail

(1) Linear Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

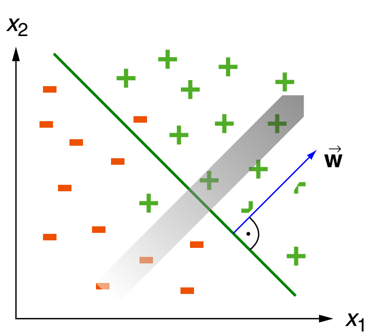
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:

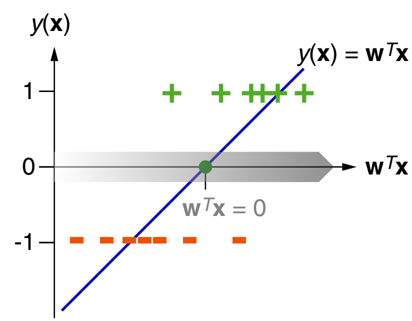
  - 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) = I_{\neq}(c, \text{sign}(y(\mathbf{x}))) = \begin{cases} 0 & \text{if } c = \text{sign}(y(\mathbf{x})) \\ 1 & \text{otherwise} \end{cases}$

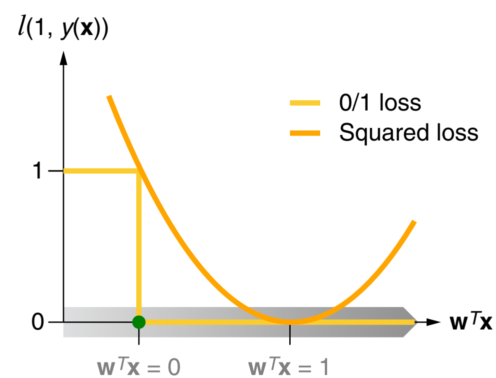
  - Squared loss. $l_2(c, y(\mathbf{x})) = (c - y(\mathbf{x}))^2$

Illustration for a particular $\mathbf{w}$:

Input space: $y()$ over hyperplane distance: Loss over hyperplane distance:







Loss Computation in Detail

(1) Linear Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

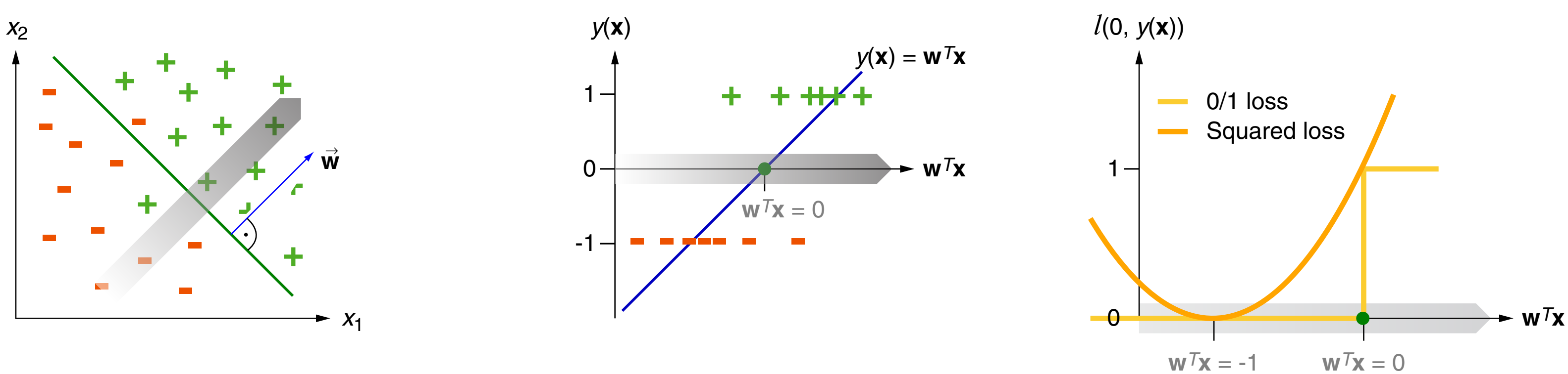
- For $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ we define the following pointwise loss functions:

  – 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) = I_{\neq}(c, \text{sign}(y(\mathbf{x}))) = \begin{cases} 0 & \text{if } c = \text{sign}(y(\mathbf{x})) \\ 1 & \text{otherwise} \end{cases}$

  – Squared loss. $l_2(c, y(\mathbf{x})) = (c - y(\mathbf{x}))^2$

Illustration for a particular $\mathbf{w}$:

Input space: $y()$ over hyperplane distance: Loss over hyperplane distance:



Remarks:

- The 0/1 loss computes the misclassification error. Recall in this regard the definition of the true misclassification rate.
- The pointwise squared loss computes the squared residual. The global squared loss, $L_2(\mathbf{w}) = \sum_{(\mathbf{x},c)\in D} l_2(c, y(\mathbf{x}))$, hence computes the residual sum of squares (RSS) related to some $\mathbf{w}$.
- $I_{\neq}$ is an indicator function that returns 1 if its arguments are *unequal* (and 0 if its arguments are equal).
- Recap. We label $y(0)$ with the "positive" class and define sign$(0) = 1$ here.
- Regarding the illustration: $\mathbf{w}^T\mathbf{x}$ is the hyperplane distance in relation to $||\mathbf{w}||$, the length of $\mathbf{w}$. By scaling $\mathbf{w}$ such that $||\mathbf{w}|| = 1$ the hyperplane distance $\mathbf{w}^T\mathbf{x}$ is normalized and is also called "geometric distance".

Loss Computation in Detail

(2) Logistic Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

- For $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ we define the following pointwise loss functions:

  – 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) = I_{\neq}(c, \lfloor y(\mathbf{x}) + 0.5 \rfloor)$ [decision rule]

  – Logistic loss. $l_\sigma(c, y(\mathbf{x})) = \begin{cases} -\log(y(\mathbf{x})) & \text{if } c = 1 \\ -\log(1 - y(\mathbf{x})) & \text{if } c = 0 \end{cases}$ [derivation]

Loss Computation in Detail

(2) Logistic Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

- For $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ we define the following pointwise loss functions:

  – 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) \;=\; I_{\neq}(c, \lfloor y(\mathbf{x}) + 0.5 \rfloor)$ [decision rule]

  – Logistic loss. $l_\sigma(c, y(\mathbf{x})) \;=\; \begin{cases} -\log(y(\mathbf{x})) & \text{if } c = 1 \\ -\log(1 - y(\mathbf{x})) & \text{if } c = 0 \end{cases}$ [derivation]

Loss Computation in Detail

(2) Logistic Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

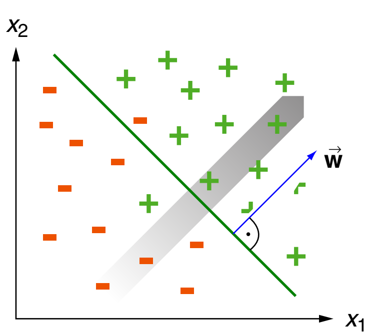
- For $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ we define the following pointwise loss functions:

  – 0/1 loss. $l_{0/1}(c, y(\mathbf{x})) \;=\; I_{\neq}(c, \lfloor y(\mathbf{x}) + 0.5 \rfloor)$ [decision rule]

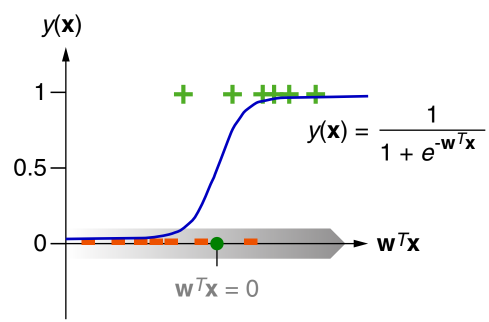
  – Logistic loss. $l_\sigma(c, y(\mathbf{x})) \;=\; \begin{cases} -\log(y(\mathbf{x})) & \text{if } c = 1 \\ -\log(1 - y(\mathbf{x})) & \text{if } c = 0 \end{cases}$ [derivation]

Illustration for a particular $\mathbf{w}$:

Input space:



$y()$ over hyperplane distance:



Loss Computation in Detail

(2) Logistic Regression

- The pointwise loss, $l(c, y(\mathbf{x}))$, quantifies the error introduced by some $\mathbf{x}$. The loss depends on a hypothesis $y()$ and the true class, c, of $\mathbf{x}$.

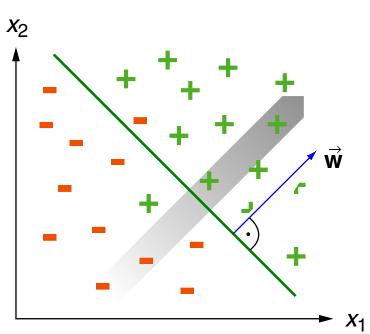
- For $y(\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ we define the following pointwise loss functions:

  – 0/1 loss. $\quad l_{0/1}(c, y(\mathbf{x})) \;=\; I_{\neq}(c, \lfloor y(\mathbf{x}) + 0.5 \rfloor)$ [decision rule]

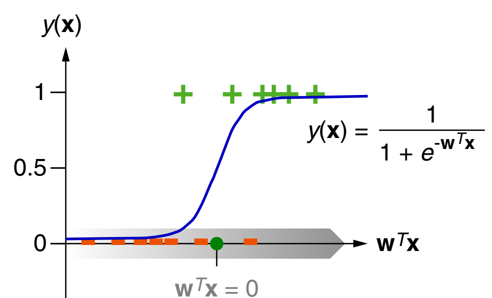
  – Logistic loss. $\quad l_{\sigma}(c, y(\mathbf{x})) \;=\; \begin{cases} -\log(y(\mathbf{x})) & \text{if } c = 1 \\ -\log(1 - y(\mathbf{x})) & \text{if } c = 0 \end{cases}$ [derivation]

Illustration for a particular $\mathbf{w}$:

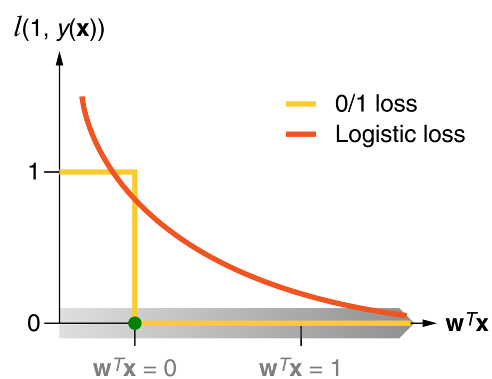
Input space:



$y()$ over hyperplane distance:



Loss over hyperplane distance:



Remarks:

- As before, the 0/1 loss computes the misclassification error.

- The pointwise logistic loss can be rewritten by combining the two cases algebraically:

$$l_\sigma(c, y(\mathbf{x})) \;=\; -c \cdot \log(y(\mathbf{x})) - (1-c) \cdot \log(1 - y(\mathbf{x}))$$

  $L_\sigma(\mathbf{w}) = \sum_{(\mathbf{x},c)\in D} l_\sigma(c, y(\mathbf{x}))$ computes the global logistic loss related to some $\mathbf{w}$.

- Recall from the derivation of the logistic loss $L_\sigma(\mathbf{w})$ that its minimization determines $\mathbf{w}_{\text{ML}}$, the most probable hypothesis in $\mathbf{R}^{p+1}$ under the logistic regression model.

- Recap. $I_{\neq}$ is an indicator function that returns 1 if its arguments are *unequal* (and 0 if its arguments are equal).

- Recap. We label $y(0)$ with the "positive" class, which is consistent with the definition of the floor function, since $\lfloor y(0) + 0.5 \rfloor = 1$.

- Q. Why not use mean squared error (MSE) as a loss function for logistic regression?
  A. MSE is not convex as loss function for logistic regression. [ai.stackexchange] [Varma 2018] [Nabert 2021] [DeLuca 2023]

