Chapter ML:II (continued)

II. Machine Learning Basics

- Rule-Based Learning of Simple Concepts
- From Regression to Classification
- Evaluating Effectiveness

From Regression to Classification

Regression versus Classification

- X is a multiset of p-dimensional feature vectors:

Customer 1	
house owner	yes
income (p.a.)	51 000 EUR
repayment (p.m.)	1 000 EUR
credit period	7 years
SCHUFA entry	no
age	37
married	yes
...	

...

Customer n	
house owner	no
income (p.a.)	55 000 EUR
repayment (p.m.)	1 200 EUR
credit period	8 years
SCHUFA entry	no
age	?
married	yes
...	

From Regression to Classification

Regression versus Classification (continued)

- X is a multiset of p-dimensional feature vectors:

Customer 1	
house owner	yes
income (p.a.)	51 000 EUR
repayment (p.m.)	1 000 EUR
credit period	7 years
SCHUFA entry	no
age	37
married	yes
...	

...

Customer n	
house owner	no
income (p.a.)	55 000 EUR
repayment (p.m.)	1 200 EUR
credit period	8 years
SCHUFA entry	no
age	?
married	yes
...	

Regression setting:

- $\mathbf{R}$ is the range of the regression function.
- y_i is the ground truth of the credit line value for $\mathbf{x}_i$, $\mathbf{x}_i \in X$.
- $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \subseteq X \times \mathbf{R}$ is a multiset of examples.

From Regression to Classification

Regression versus Classification (continued)

- X is a multiset of p-dimensional feature vectors:

Customer 1	
house owner	yes
income (p.a.)	51 000 EUR
repayment (p.m.)	1 000 EUR
credit period	7 years
SCHUFA entry	no
age	37
married	yes
...	

...

Customer n	
house owner	no
income (p.a.)	55 000 EUR
repayment (p.m.)	1 200 EUR
credit period	8 years
SCHUFA entry	no
age	?
married	yes
...	

Regression setting:

- $\mathbf{R}$ is the range of the regression function.
- y_i is the ground truth of the credit line value for $\mathbf{x}_i$, $\mathbf{x}_i \in X$.
- $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \subseteq X \times \mathbf{R}$ is a multiset of examples.

Classification setting:

- $C = \{-1, 1\}$ is a set of two classes. Similarly: $\{0, 1\}$, $\{\ominus, \oplus\}$, $\{\text{no}, \text{yes}\}$, etc.
- c_i is the ground truth of the creditworthiness class for $\mathbf{x}_i$, $\mathbf{x}_i \in X$.
- $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_n, c_n)\} \subseteq X \times C$ is a multiset of examples.

From Regression to Classification

The Linear Regression Model

- Given $\mathbf{x}$, predict a real-valued output under a linear model function:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot x_j$$

- Vector notation with $x_0 = 1$ and $\mathbf{w} = (w_0, w_1, \ldots, w_p)^T$:

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

From Regression to Classification

The Linear Regression Model (continued)

- Given $\mathbf{x}$, predict a real-valued output under a linear model function:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot x_j$$

- Vector notation with $x_0 = 1$ and $\mathbf{w} = (w_0, w_1, \ldots, w_p)^T$:

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

- Given $\mathbf{x}_1, \ldots, \mathbf{x}_n$, assess goodness of fit as residual sum of squares:

$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - y(\mathbf{x}_i))^2 = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \tag{1}$$

From Regression to Classification

The Linear Regression Model (continued)

- Given $\mathbf{x}$, predict a real-valued output under a linear model function:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot x_j$$

- Vector notation with $x_0 = 1$ and $\mathbf{w} = (w_0, w_1, \ldots, w_p)^T$:

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

- Given $\mathbf{x}_1, \ldots, \mathbf{x}_n$, assess goodness of fit as residual sum of squares:

$$\text{RSS}(\mathbf{w}) \;=\; \sum_{i=1}^{n} (y_i - y(\mathbf{x}_i))^2 \;=\; \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \tag{1}$$

- Estimate optimum $\mathbf{w}$ by minimizing the residual sum of squares:

$$\hat{\mathbf{w}} \;=\; \underset{\mathbf{w} \in \mathbf{R}^{p+1}}{\text{argmin}} \; \text{RSS}(\mathbf{w}) \tag{2}$$

Remarks (residuals):

- A *residual* is the difference between a target value (ground truth, observation) y_i and the estimated value $y(\mathbf{x}_i)$ of the model function.
- The residual sum of squares, RSS, is the sum of squares of the residuals. It is also known as the sum of squared residuals, SSR, or the sum of squared errors of estimates, SSE.
- The RSS term quantifies the regression error—or similarly, the goodness of fit—in the form of a single value.
- RSS provides several numerical and theoretical advantages, but it is not the only possibility to assess the goodness of fit (= error) between observed values and the model function. Alternative approaches for quantifying the error include absolute residual values or likelihood estimates.
- The error computation is also called loss computation, cost computation, or generally, performance computation. Similarly, for the right-hand side of Equation (1) the following names are used: error function, loss function, cost function, or generally, performance term.

  Measures that quantify this kind of performance are called *effectiveness* measures. This term must not be confused with *efficiency* measures, which quantify the computational effort or runtime performance of a method.
- Residual $\neq$ Loss. Observe the subtle difference between the two concepts “residual” and “loss” (similarly: “error”, “cost”). The former denotes the difference between a target value (ground truth, observation) and its estimate, whereas the latter denotes the *interpretation of this difference*.

Remarks (randomness and distributions):

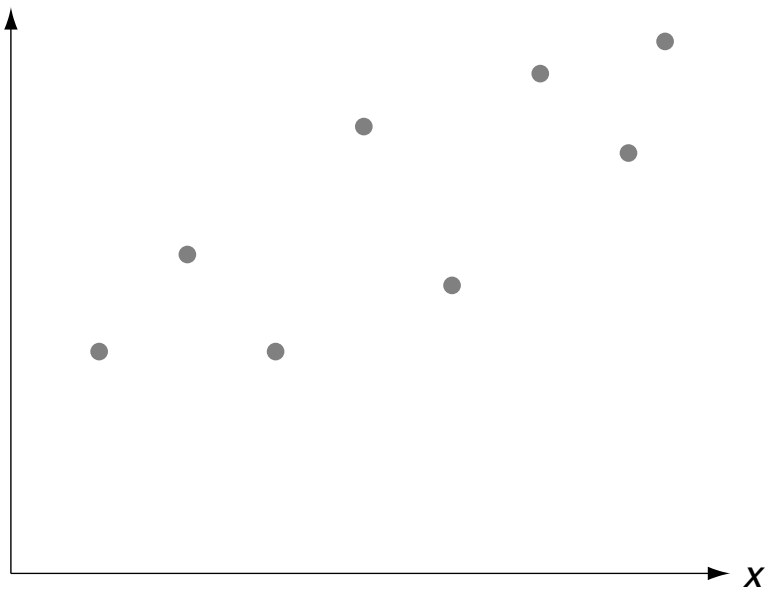
- The y_i are considered as realizations of n respective random variables Y_i. Btw., do not confuse the function $y()$ with a realization y_i.
- Equation (2): Estimating $\hat{\mathbf{w}}$ by RSS minimization is based on the following assumptions:
  1. The probability distributions of the Y_i have the same variance.
  2. The expectations $E[Y_i]$ of the Y_i lie on a straight line, known as the true (population) regression line: $E[Y_i] = \mathbf{w}^{*T}\mathbf{x}_i$. I.e., the relation between the $\mathbf{x}_i$ and the observed y_i can be explained completely by a linear model function.
  3. The random variables Y_i are statistically independent.

  These assumptions are called the *weak set* (of assumptions). Along with a fourth assumption about the distribution shape of Y_i they become the *strong set* of assumptions.
- Y_i may also be defined as $y(\mathbf{x}_i) + E_i$, in which case the *disturbance term* E_i has the same distribution as Y_i but the mean 0 (while Y_i has the mean $\mathbf{w}^T\mathbf{x}_i$).
- Within the classical regression setting the variable $\mathbf{x}$, also called regressor, is a controlled variable. I.e., its instances $\mathbf{x}_i$, $i = 1, \ldots, n$, are not considered as outcomes of a random experiment: the $\mathbf{x}_i$ are given, chosen with intent, or constructed without any effect of chance.

  Within the typical machine learning setting, the occurrence of feature vectors—more general, the sample formation process underlying X—is governed by a probability distribution: certain observations may be more likely than others, and hence each feature vector $\mathbf{x}_i$ is considered as the realization of a random vector $\mathbf{X}_i$.

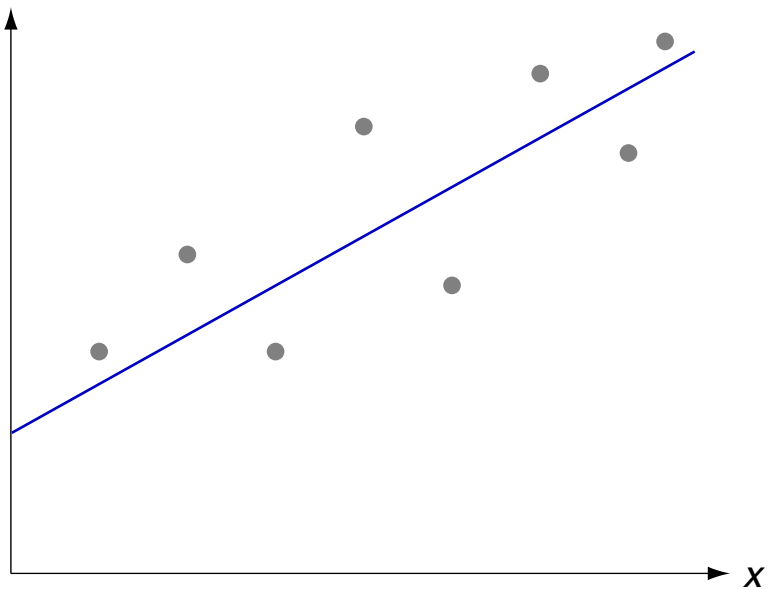
From Regression to Classification

One-Dimensional Feature Space



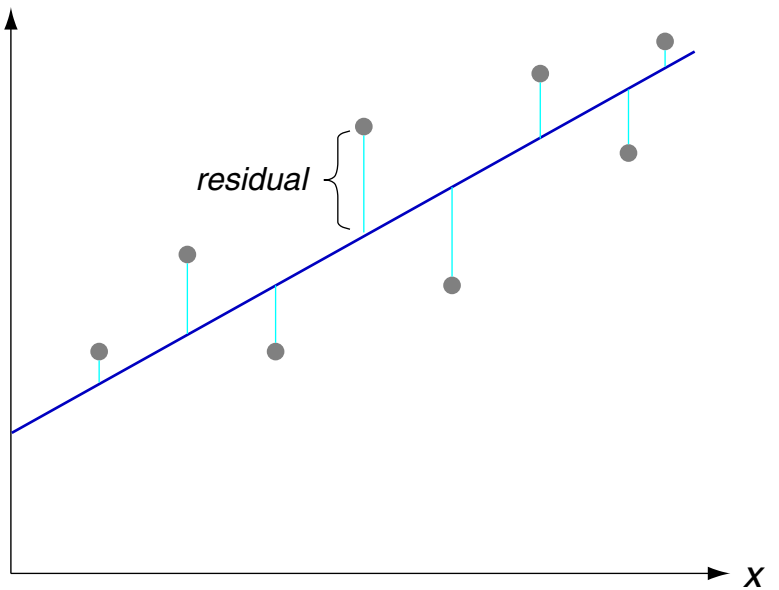
From Regression to Classification

One-Dimensional Feature Space (continued)



From Regression to Classification

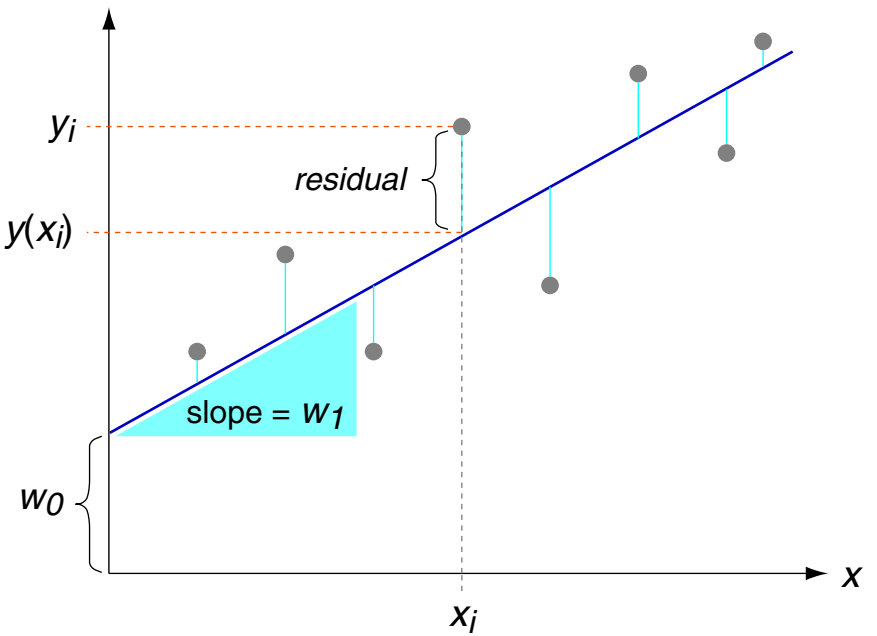
One-Dimensional Feature Space (continued)



$$\text{RSS} = \sum_{i=1}^{n} (y_i - y(x_i))^2$$

From Regression to Classification

One-Dimensional Feature Space (continued)



$$y(x) = w_0 + w_1 \cdot x, \qquad \text{RSS}(w_0, w_1) = \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2$$

From Regression to Classification

One-Dimensional Feature Space (continued) [higher-dimensional]

Minimize RSS(w_0, w_1) via a direct method:

1. $$\frac{\partial}{\partial w_0} \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2 \;=\; 0$$

$$\rightsquigarrow \;\ldots\; \rightsquigarrow \quad w_0 \;=\; \frac{1}{n} \sum_{i=1}^{n} y_i \;-\; \frac{w_1}{n} \sum_{i=1}^{n} x_i \;=\; \bar{y} - w_1 \cdot \bar{x}$$

From Regression to Classification

One-Dimensional Feature Space (continued) [higher-dimensional]

Minimize RSS(w_0, w_1) via a direct method:

1. $$\frac{\partial}{\partial w_0} \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2 \;=\; 0$$

$$\rightsquigarrow \;\ldots\; \rightsquigarrow \quad w_0 \;=\; \frac{1}{n} \sum_{i=1}^{n} y_i \;-\; \frac{w_1}{n} \sum_{i=1}^{n} x_i \;=\; \bar{y} - w_1 \cdot \bar{x}$$

2. $$\frac{\partial}{\partial w_1} \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2 \;=\; 0$$

$$\rightsquigarrow \;\ldots\; \rightsquigarrow \qquad w_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

From Regression to Classification

One-Dimensional Feature Space (continued) [higher-dimensional]

Minimize RSS(w_0, w_1) via a direct method:

1. $$\frac{\partial}{\partial w_0} \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2 \;=\; 0$$

$$\rightsquigarrow \;\ldots\; \rightsquigarrow \quad \hat{w}_0 \;=\; \frac{1}{n} \sum_{i=1}^{n} y_i \;-\; \frac{w_1}{n} \sum_{i=1}^{n} x_i \;=\; \bar{y} - \hat{w}_1 \cdot \bar{x}$$

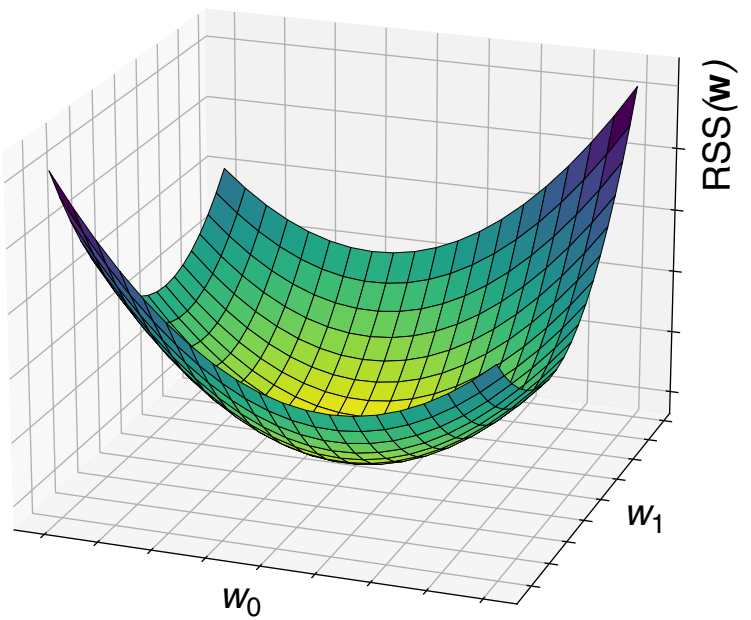
2. $$\frac{\partial}{\partial w_1} \sum_{i=1}^{n} (y_i - w_0 - w_1 \cdot x_i)^2 \;=\; 0$$

$$\rightsquigarrow \;\ldots\; \rightsquigarrow \quad \hat{w}_1 \equiv w_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

From Regression to Classification

One-Dimensional Feature Space (continued)

Illustration of the task of minimizing $\text{RSS}(\mathbf{w}) = \sum_{i=1}^{n}(y_i - \mathbf{w}^T\mathbf{x}_i)^2$.



From Regression to Classification

Higher-Dimensional Feature Space

- Recall Equation (1) :

$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

- Let X denote the $n \times (p+1)$ matrix where row i is the extended input vector $(1 \;\; \mathbf{x}_i^T)$ with $(\mathbf{x}_i, y_i) \in D$.

  Let $\mathbf{y}$ denote the n-vector of target values y_i in the training set D.

From Regression to Classification

Higher-Dimensional Feature Space (continued)

- Recall Equation (1) :

$$\text{RSS}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

- Let X denote the $n \times (p+1)$ matrix where row i is the extended input vector $(1 \;\; \mathbf{x}_i^T)$ with $(\mathbf{x}_i, y_i) \in D$.

  Let $\mathbf{y}$ denote the n-vector of target values y_i in the training set D.

$\rightsquigarrow$ $\text{RSS}(\mathbf{w}) = (\mathbf{y} - X\mathbf{w})^T (\mathbf{y} - X\mathbf{w})$

RSS$(\mathbf{w})$ is a quadratic function in $p+1$ parameters.

From Regression to Classification

Higher-Dimensional Feature Space (continued) [one-dimensional]

Minimize $\mathrm{RSS}(\mathbf{w})$ via a direct method:

$$\frac{\partial\,\mathrm{RSS}}{\partial\mathbf{w}} = -2X^T(\mathbf{y}-X\mathbf{w}) = 0, \qquad \frac{\partial^2\,\mathrm{RSS}}{\partial\mathbf{w}\partial\mathbf{w}^T} = -2X^TX$$ [Wikipedia 1, 2, 3]

$$\begin{aligned} X^T(\mathbf{y}-X\mathbf{w}) &= 0 \\ \Leftrightarrow \quad X^TX\mathbf{w} &= X^T\mathbf{y} \\ \rightsquigarrow \quad \mathbf{w} &= (X^TX)^{-1}X^T\,\mathbf{y} \end{aligned}$$

From Regression to Classification

Higher-Dimensional Feature Space (continued) [one-dimensional]

Minimize $\mathrm{RSS}(\mathbf{w})$ via a direct method:

$$\frac{\partial \mathrm{RSS}}{\partial \mathbf{w}} = -2X^T(\mathbf{y} - X\mathbf{w}) = 0, \qquad \frac{\partial^2 \mathrm{RSS}}{\partial \mathbf{w} \partial \mathbf{w}^T} = -2X^T X$$ [Wikipedia 1, 2, 3]

$$\begin{aligned} X^T(\mathbf{y} - X\mathbf{w}) &= 0 \\ \Leftrightarrow \quad X^T X\mathbf{w} &= X^T\mathbf{y} \\ \rightsquigarrow \quad \mathbf{w} &= \underbrace{(X^T X)^{-1} X^T}\, \mathbf{y} \end{aligned}$$

Normal equations. [Mathworld]

If X has full column rank $p+1$.

Pseudoinverse of X [Wikipedia]

From Regression to Classification

Higher-Dimensional Feature Space (continued) [one-dimensional]

Minimize $\mathrm{RSS}(\mathbf{w})$ via a direct method:

$$\frac{\partial\,\mathrm{RSS}}{\partial \mathbf{w}} = -2X^T(\mathbf{y} - X\mathbf{w}) = 0, \qquad \frac{\partial^2\,\mathrm{RSS}}{\partial \mathbf{w}\partial \mathbf{w}^T} = -2X^TX$$ [Wikipedia 1, 2, 3]

$$\begin{aligned} X^T(\mathbf{y} - X\mathbf{w}) &= 0 \\ \Leftrightarrow \quad X^TX\mathbf{w} &= X^T\mathbf{y} \qquad \text{Normal equations. [Mathworld]} \\ \rightsquigarrow \quad \hat{\mathbf{w}} \equiv \mathbf{w} &= \underbrace{(X^TX)^{-1}X^T}\,\mathbf{y} \qquad \text{If } X \text{ has full column rank } p{+}1. \end{aligned}$$

Pseudoinverse of X [Wikipedia]

$$\hat{y}(\mathbf{x}_i) = \hat{\mathbf{w}}^T\mathbf{x}_i$$ Regression function with least squares estimator $\hat{\mathbf{w}}$.

$$\begin{aligned} \hat{\mathbf{y}} &= X\hat{\mathbf{w}} \qquad \text{The } n\text{-vector of fitted values at the training input.} \\ &= X(X^TX)^{-1}X^T\mathbf{y} \end{aligned}$$

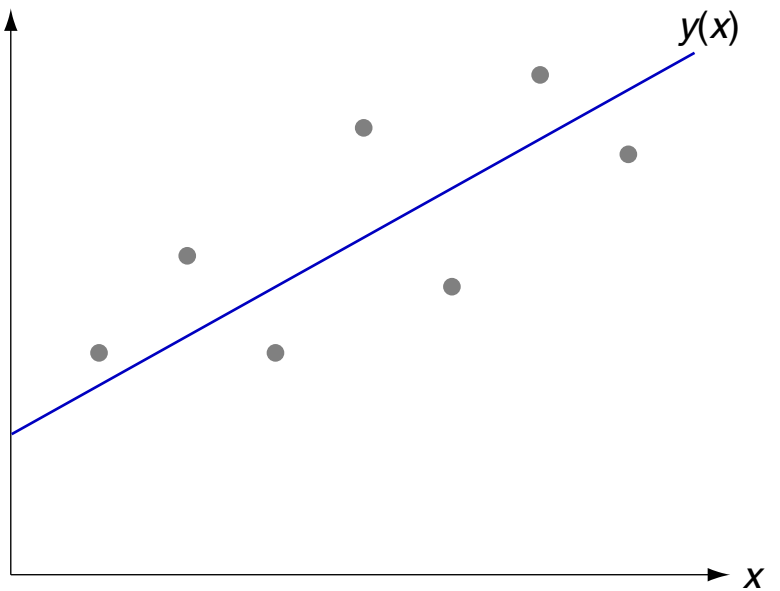
Remarks:

- A curve fitting (or regression) method that is based on the minimization of squared residuals is called a *method of least squares*.
- Various approaches for operationalizing the method of least squares have been devised, in particular for the case of linear model functions. From a numerical viewpoint one can distinguish iterative methods, such as the LMS algorithm, and direct methods, such as solving the normal equations via computing the pseudoinverse.
- More on direct methods. While solving the normal equations is usually fast, it suffers from several deficits: it is numerically unstable and requires singularity handling. Numerically more stable and more accurate methods are based on the QR decomposition and the singular value decomposition, SVD.
- QR decomposition can deal with problems of up to 10^4 variables, provided a dense problem structure. For significantly larger problems (additional 1-2 orders of magnitudes) as well as for sparse matrices iterative solvers are the choice. Even larger, dense problems may be tackled with Artificial Neural Networks.

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 1$)

Regression learns a real-valued function given as $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.



$$y(x) = (w_0 \;\; w_1)\begin{pmatrix}1\\ x\end{pmatrix}$$

From Regression to Classification

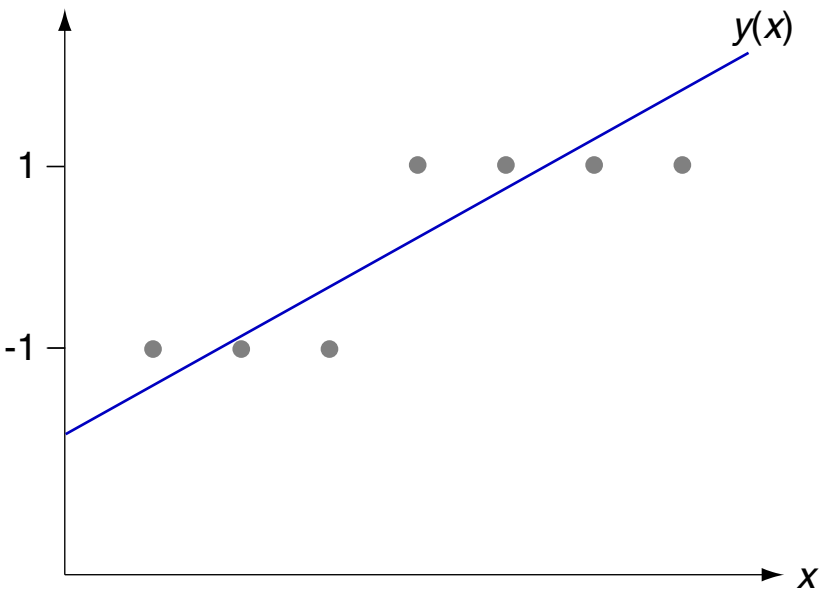
Linear Regression for Classification (illustrated for $p = 1$) (continued)

Binary-valued (± 1) functions are also real-valued.

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 1$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.

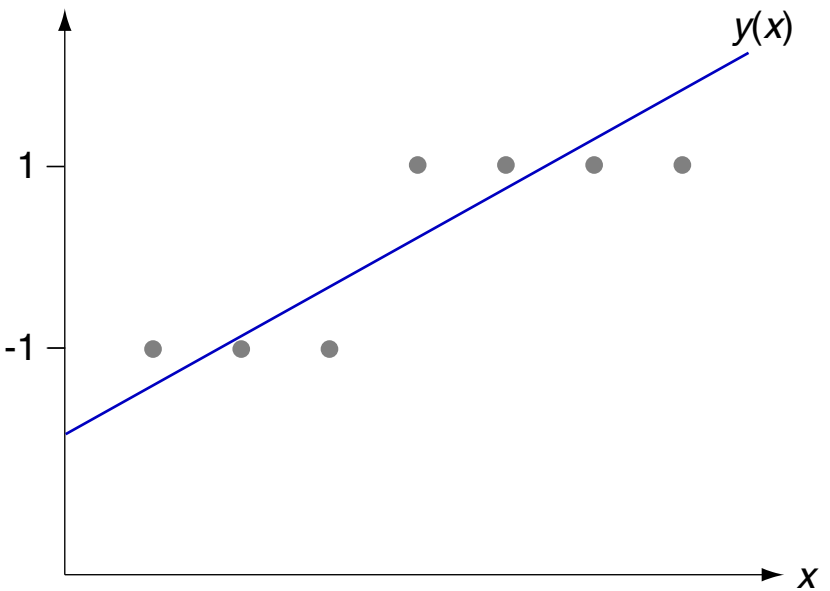


$$y(x) = (w_0 \;\; w_1) \begin{pmatrix} 1 \\ x \end{pmatrix}$$

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 1$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.



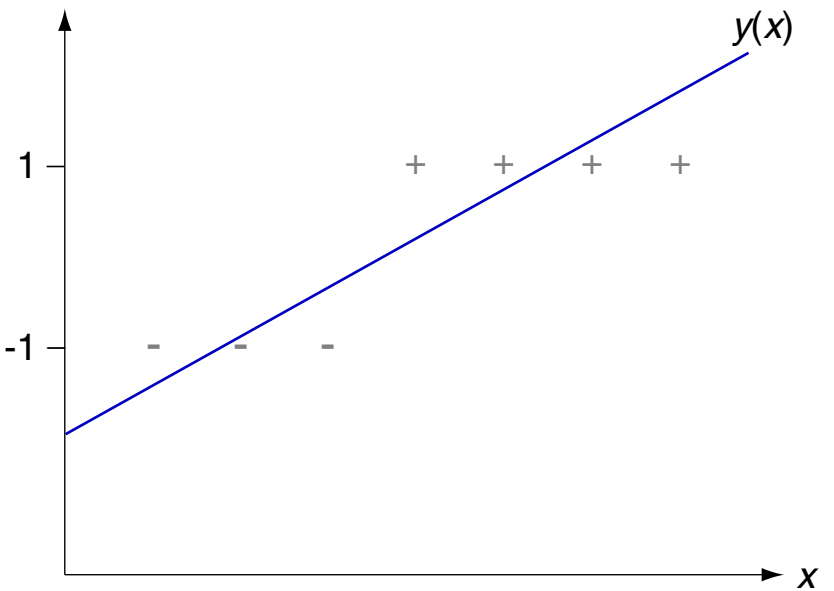
The function "sign$(\mathbf{w}^T \mathbf{x}_i)$" is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 1$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.



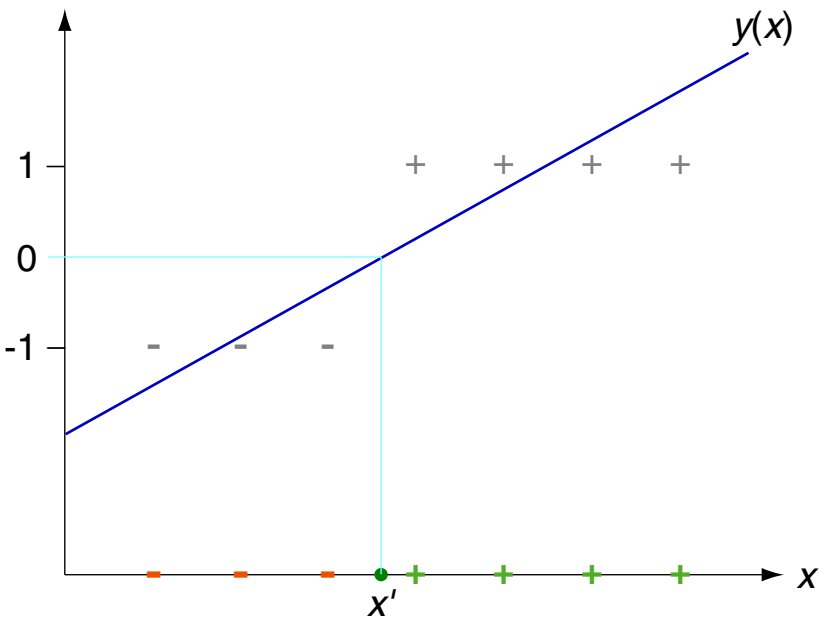
The function "sign$(\mathbf{w}^T \mathbf{x}_i)$" is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 1$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.



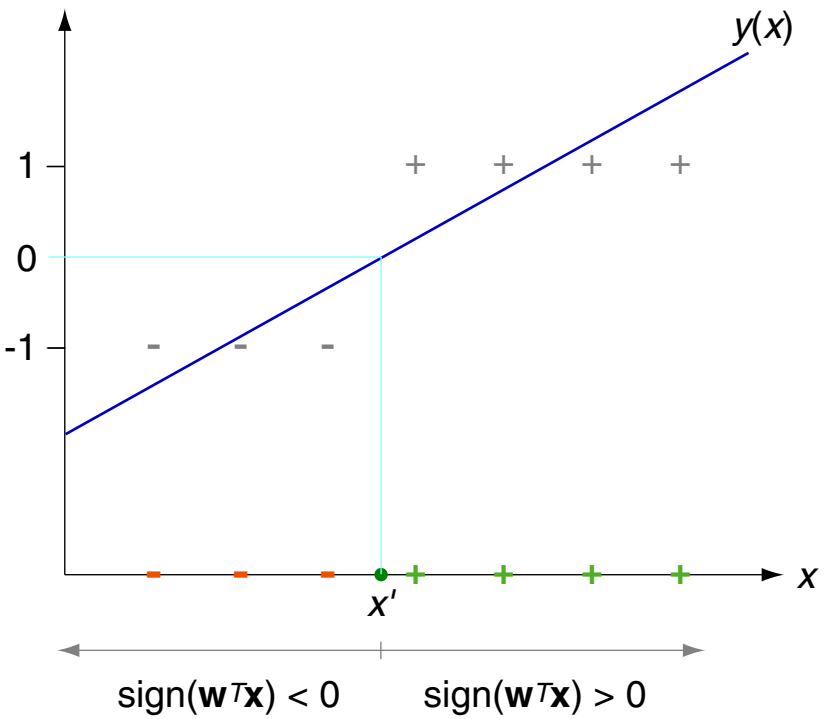
The function "sign$(\mathbf{w}^T \mathbf{x}_i)$" is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 1$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.



- The discrimination point, •, is defined by that x that fulfills $w_0 + w_1 \cdot x = 0$.
- For $p = 2$ we are given a discrimination *line*.

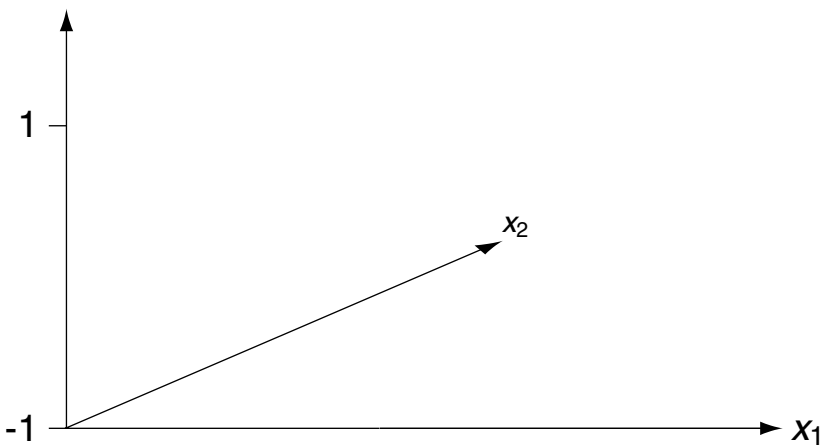
Remarks:

($\star$) 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 $x_0 = 1$.

- The sign function is three-valued, with $\text{sign}(z) = -1$ (0, 1) for $z < 0$ ($z = 0$, $z > 0$)—i.e., the case with $\mathbf{w}^T\mathbf{x} = 0$ needs special treatment. Without loss of generality we will label $y(0)$ with the "positive" class (1, $\oplus$, yes, etc.) and define $\text{sign}(0) = 1$ in the respective algebraic expressions.

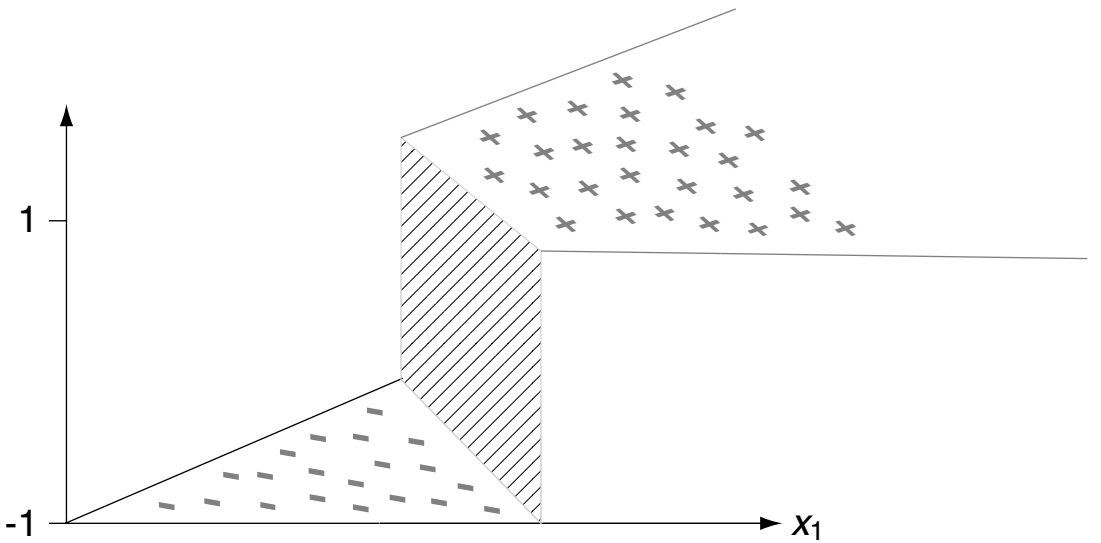
From Regression to Classification

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



From Regression to Classification

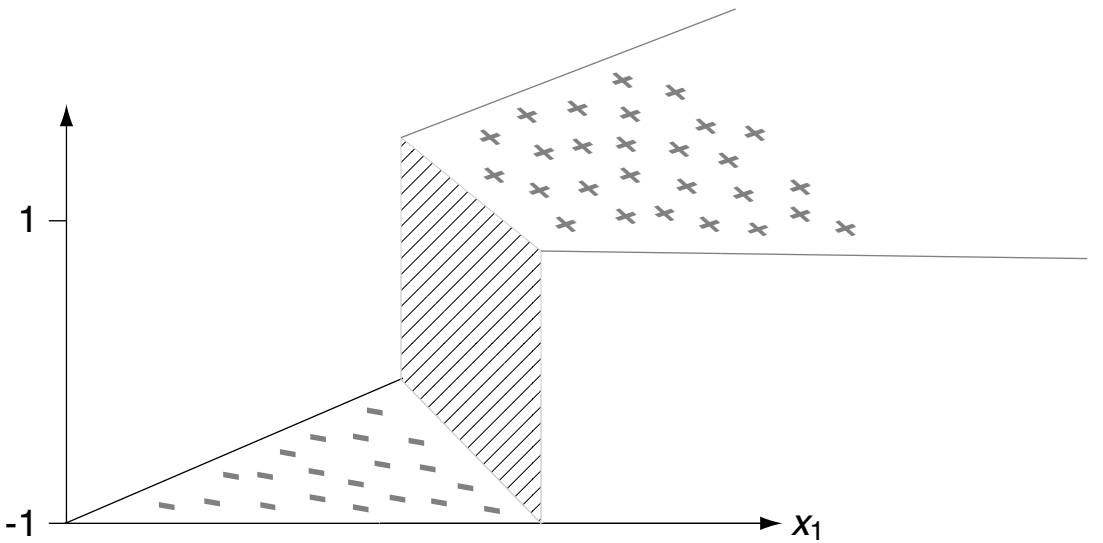
Linear Regression for Classification (illustrated for $p = 2$) (continued)



From Regression to Classification

Linear Regression for Classification (illustrated for $p = 2$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.

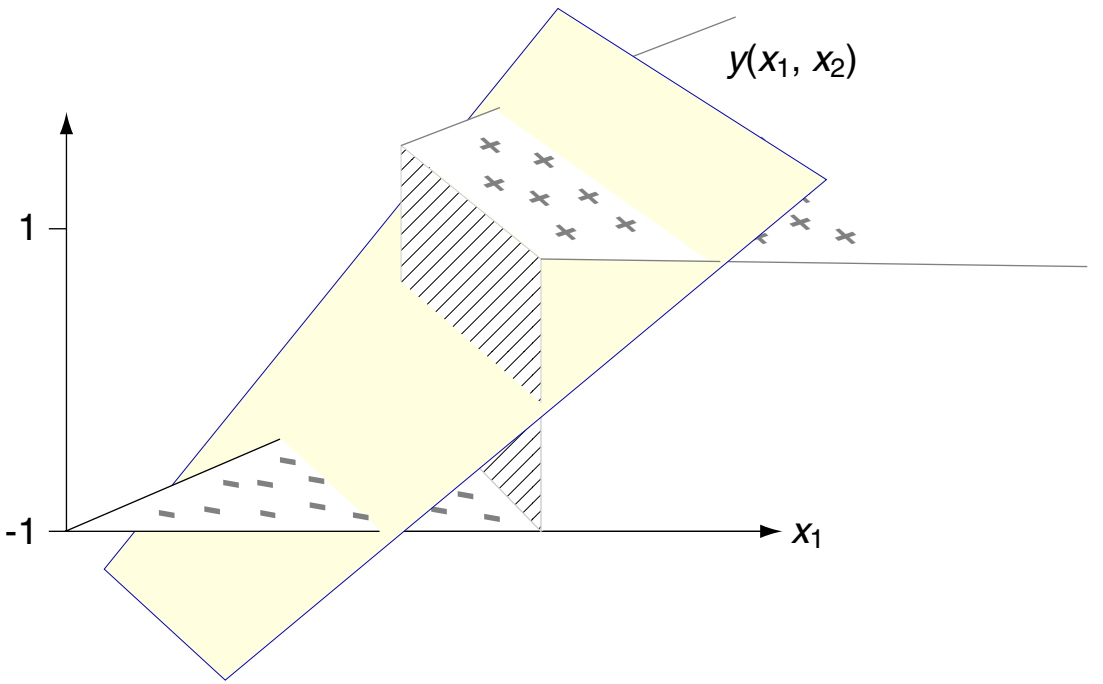


$$y(x_1, x_2) = (w_0 \;\; w_1 \;\; w_2) \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix}$$

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 2$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.



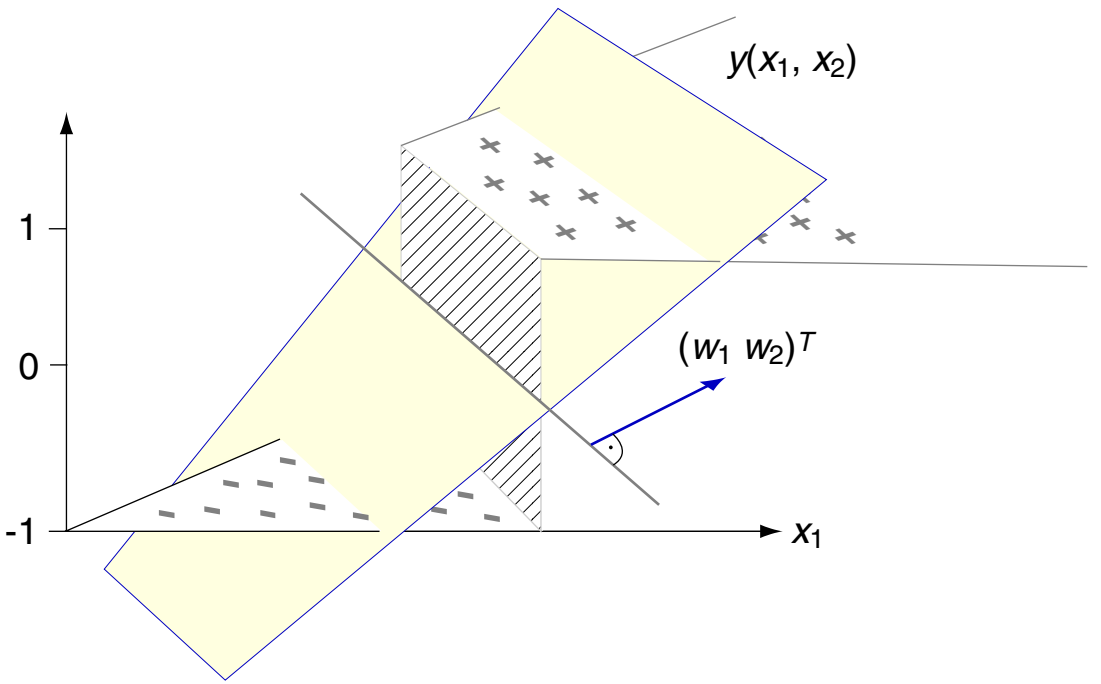
The function "sign$(\mathbf{w}^T \mathbf{x}_i)$" is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 2$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.



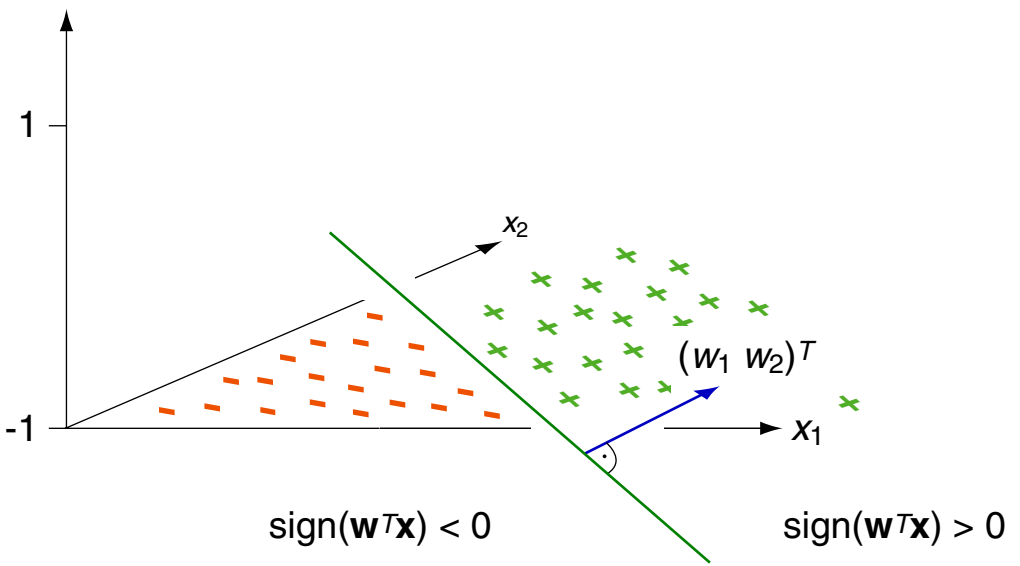
The function "sign$(\mathbf{w}^T \mathbf{x}_i)$" is likely to agree with $y_i = \pm 1$.

- Regression: $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$
- Classification: $y(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x})$

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 2$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.

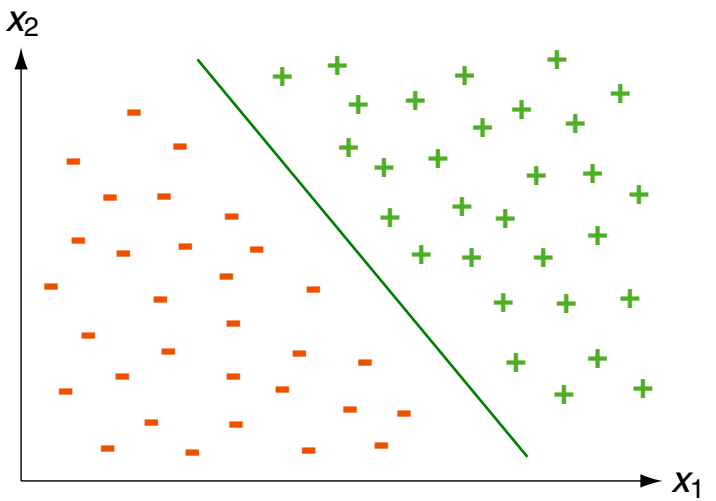


- The discrimination line, —, is defined by the $\mathbf{x}$ that fulfill $w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0$.
- For $p = 3$ $(p > 3)$ we are given a discriminating *(hyper)plane*.

From Regression to Classification

Linear Regression for Classification (illustrated for $p = 2$) (continued)

Use linear regression to learn $\mathbf{w}$ from D, where $y_i = \pm 1 \approx y(\mathbf{x}_i) \overset{(\star)}{=} \mathbf{w}^T \mathbf{x}_i$.



- The discrimination line, —, is defined by the $\mathbf{x}$ that fulfill $w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0$.
- For $p = 3$ $(p > 3)$ we are given a discriminating *(hyper)plane*.

Remarks:

- The shown figures illustrate how (linear) regression methods that are applied in the input-output space implicitly define a hyperplane in the input space.

  In general, linear regression is not the best choice to solve classification problems: imbalanced class distributions and outliers can severely impair the classification effectiveness.

- A suited regression method for classification is logistic regression, introduced in the part Linear Models, which estimates the probability of class membership. Note that also logistic regression is a linear classifier since its encoded hypothesis is a linear function in the parameters $\mathbf{w}$.

  An illustration of the input-output space of the logistic regression model along with the implicitly defined hyperplane is shown here.

From Regression to Classification

Linear Model Function Variants

The components (features) of the input vector $\mathbf{x} = (x_1, \ldots, x_p)$ can stem from different sources [Hastie et al. 2001]:

1. quantitative inputs
2. transformations of quantitative inputs, such as $\log x_j$, $\sqrt{x_j}$
3. basis expansions, such as $x_j = (x_1)^j$
4. encoding of a qualitative variable g, $g \in \{1, \ldots, p\}$, as $x_j = \mathcal{I}(g \doteq j)$
5. interactions between variables, such as $x_3 = x_1 \cdot x_2$

From Regression to Classification

Linear Model Function Variants (continued)

The components (features) of the input vector $\mathbf{x} = (x_1, \ldots, x_p)$ can stem from different sources [Hastie et al. 2001]:

1. quantitative inputs
2. transformations of quantitative inputs, such as $\log x_j$, $\sqrt{x_j}$
3. basis expansions, such as $x_j = (x_1)^j$
4. encoding of a qualitative variable g, $g \in \{1, \ldots, p\}$, as $x_j = \mathcal{I}(g \doteq j)$
5. interactions between variables, such as $x_3 = x_1 \cdot x_2$

No matter the source of the x_j, the model is still linear in its parameters $\mathbf{w}$:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot \phi_j(x_j)$$

From Regression to Classification

Linear Model Function Variants (continued)

The components (features) of the input vector $\mathbf{x} = (x_1, \ldots, x_p)$ can stem from different sources [Hastie et al. 2001]:

1. quantitative inputs
2. transformations of quantitative inputs, such as $\log x_j$, $\sqrt{x_j}$
3. basis expansions, such as $x_j = (x_1)^j$
4. encoding of a qualitative variable g, $g \in \{1, \ldots, p\}$, as $x_j = \mathcal{I}(g \doteq j)$
5. interactions between variables, such as $x_3 = x_1 \cdot x_2$

No matter the source of the x_j, the model is still linear in its parameters $\mathbf{w}$:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot \phi_j(x_j)$$

- linear in the parameters: $y(\mathbf{w})$ is a linear function

From Regression to Classification

Linear Model Function Variants (continued)

The components (features) of the input vector $\mathbf{x} = (x_1, \ldots, x_p)$ can stem from different sources [Hastie et al. 2001]:

1. quantitative inputs
2. transformations of quantitative inputs, such as $\log x_j$, $\sqrt{x_j}$
3. basis expansions, such as $x_j = (x_1)^j$
4. encoding of a qualitative variable g, $g \in \{1, \ldots, p\}$, as $x_j = \mathcal{I}(g \doteq j)$
5. interactions between variables, such as $x_3 = x_1 \cdot x_2$

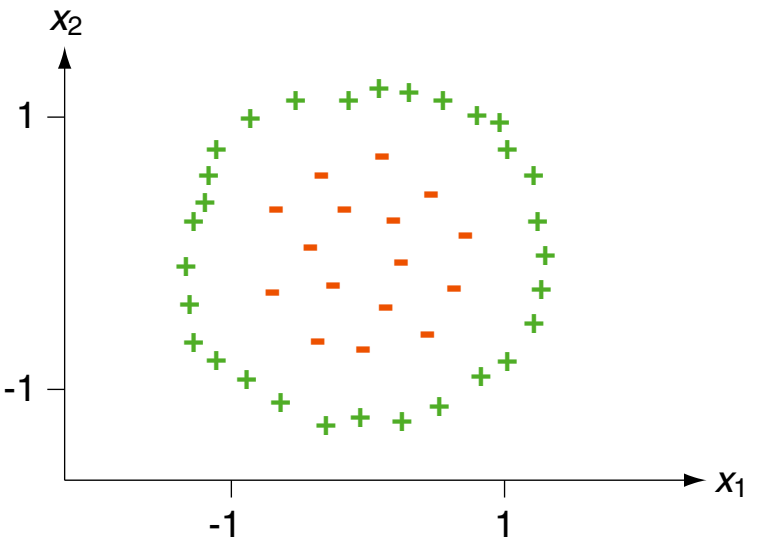
No matter the source of the x_j, the model is still linear in its parameters $\mathbf{w}$:

$$y(\mathbf{x}) = w_0 + \sum_{j=1}^{p} w_j \cdot \phi_j(x_j)$$

- linear in the parameters: $y(\mathbf{w})$ is a linear function
- basis functions: input variables (space) become(s) feature variables (space)

From Regression to Classification

Non-Linear Decision Boundaries [logistic regression]

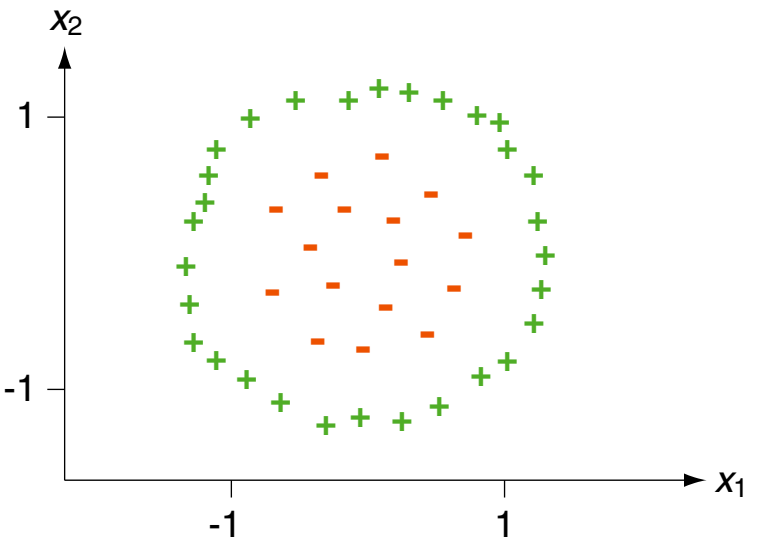


Higher order polynomial terms in the features (linear in the parameters):

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

From Regression to Classification

Non-Linear Decision Boundaries (continued) [logistic regression]

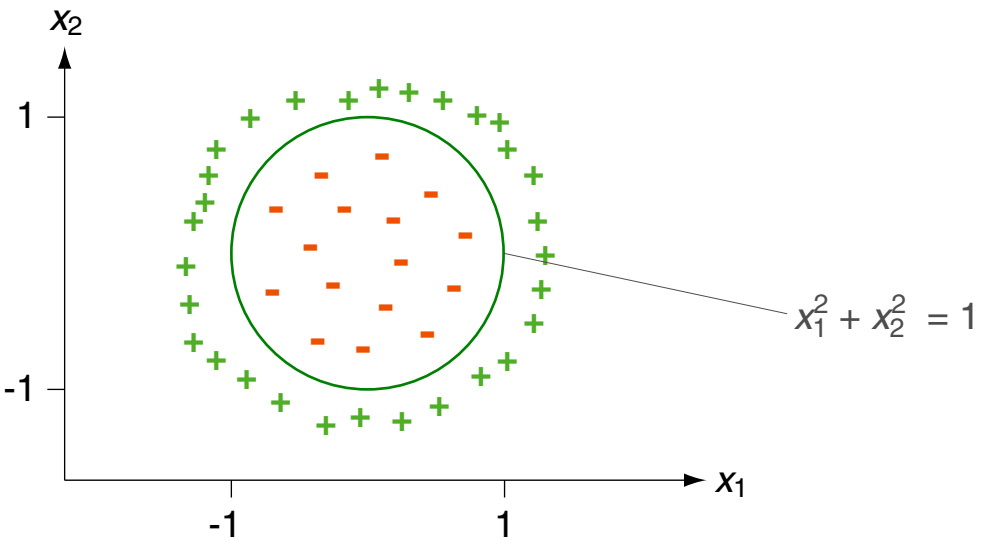


Higher order polynomial terms in the features (linear in the parameters):

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

From Regression to Classification

Non-Linear Decision Boundaries (continued) [logistic regression]



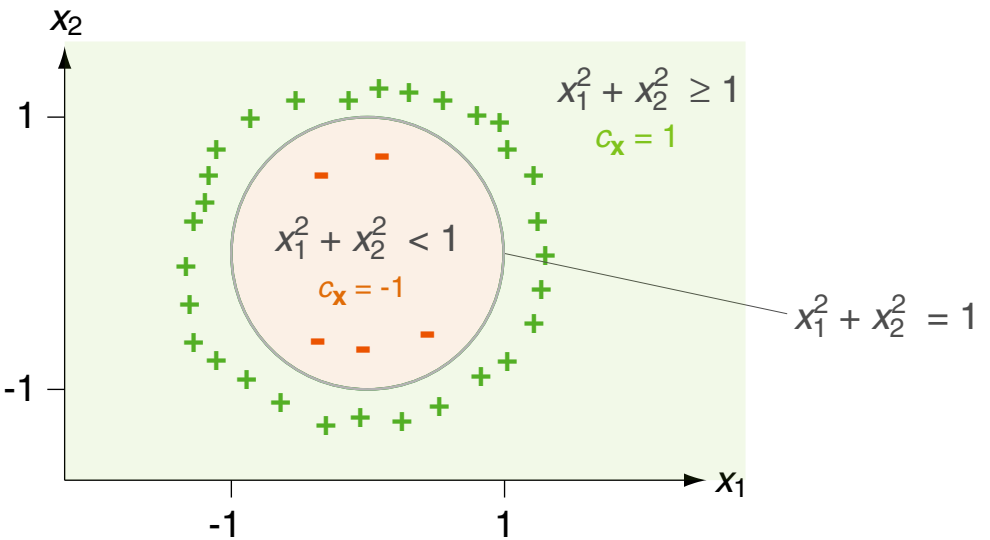
Higher order polynomial terms in the features (linear in the parameters):

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

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

From Regression to Classification

Non-Linear Decision Boundaries (continued) [logistic regression]



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

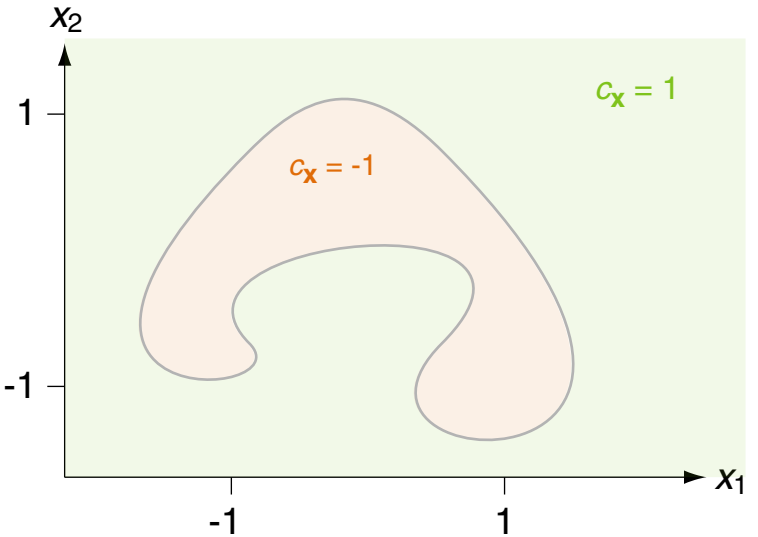
$$y(\mathbf{x}) = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_1^2 + w_4 \cdot x_2^2$$

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

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

From Regression to Classification

Non-Linear Decision Boundaries (continued) [logistic regression]



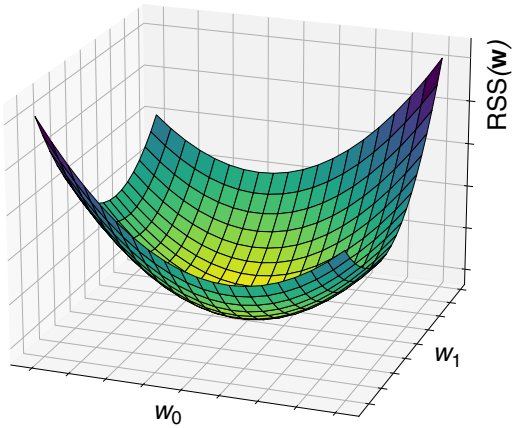
More complex polynomials entail more complex decision boundaries:

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

From Regression to Classification

Methods of Least Squares: Iterative versus Direct Methods

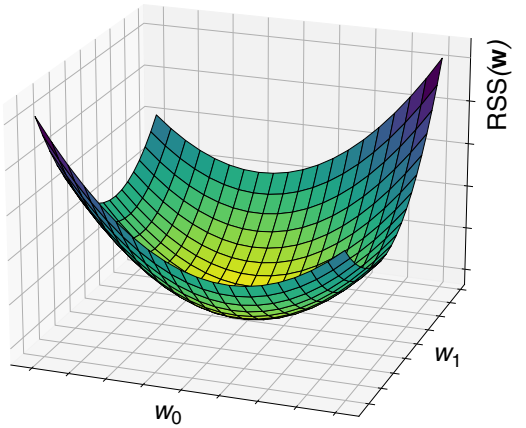
$$\operatorname*{argmin}_{\mathbf{w}} \text{RSS}(\mathbf{w}), \quad \text{with } \text{RSS}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$



From Regression to Classification

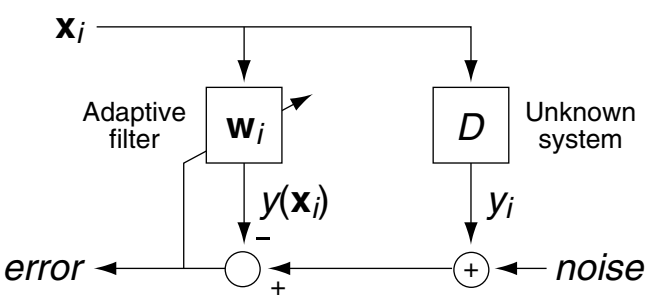
Methods of Least Squares: Iterative versus Direct Methods (continued)

$$\underset{\mathbf{w}}{\text{argmin}}\ \text{RSS}(\mathbf{w}), \quad \text{with}\ \text{RSS}(\mathbf{w}) = \sum_{i=1}^{n}(y_i - \mathbf{w}^T\mathbf{x}_i)^2$$



LMS algorithm:

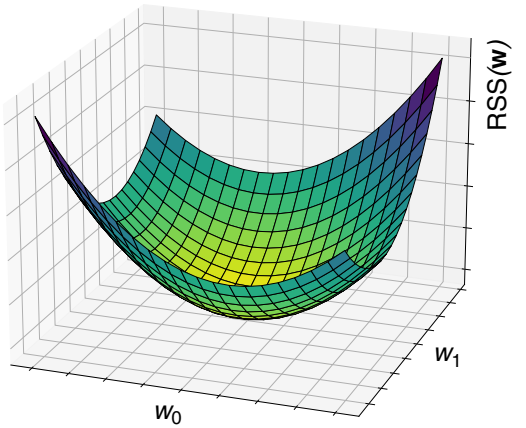
- applicable as online algorithm
- robust algorithm structure
- unsatisfactory convergence
- allows stochastic sampling



From Regression to Classification

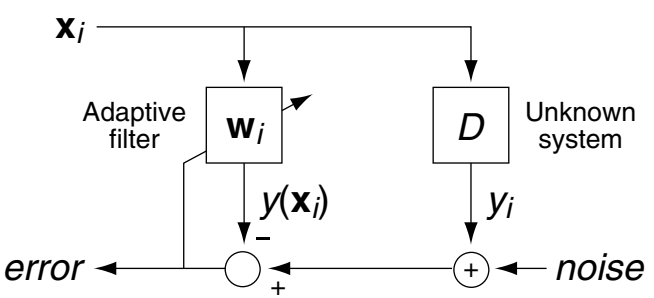
Methods of Least Squares: Iterative versus Direct Methods (continued)

$$\underset{\mathbf{w}}{\text{argmin}}\ \text{RSS}(\mathbf{w}), \quad \text{with}\ \text{RSS}(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$



LMS algorithm:

- applicable as online algorithm
- robust algorithm structure
- unsatisfactory convergence
- allows stochastic sampling



Normal equations:

- needs complete data
- numerically unstable
- requires singularity handling
- hardly applicable to big data

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y}$$

Remarks:

- The principle of RSS minimization is orthogonal to (= independent of) the type of the model function $y()$, i.e., independent of its dimensionality as well as its linearity or nonlinearity.
- To fit the parameters $\mathbf{w}$ of a (one-dimensional, multi-dimensional, linear, nonlinear) model function $y()$, both the LMS algorithm and direct methods exploit information about the derivative of the RSS term with respect to $\mathbf{w}$. I.e., even if *classification* and not regression is the goal, the distance to the decision boundary (and not the zero-one-loss) is computed, since the zero-one-loss is non-differentiable.
- For a linear model function $y()$, RSS is a convex function and hence a single, global optimum exists.
- A main goal of machine learning approaches is to avoid overfitting. Overfitting, in turn, is caused by an inadequate (too high) model function complexity—or, similarly, by insufficient data. A means to reduce the model function complexity is regularization. Both topics are treated in the part Linear Models.
- Regularization will introduce additional constraints for the model function $y()$ or the parameter vector $\mathbf{w}$. With regularization the minimization expression (2) will have two summands: a performance term such as the RSS term, and a penalizing term such as a norm. As before, the first term captures the model function's goodness depending on $\mathbf{w}$, whereas the second term restricts the absolute values of the model function's parameters $\mathbf{w}$.

From Regression to Classification

Properties of the Solution

Theorem 1 (Gauss-Markov)

Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be a multiset of examples to be fitted with a linear model function as $y(\mathbf{x}) \overset{(\star)}{=} \mathbf{x}^T\mathbf{w}$. Within the class of linear unbiased estimators for $\mathbf{w}$, the least squares estimator $\hat{\mathbf{w}}$ has minimum variance, i.e., is most efficient.

From Regression to Classification

Properties of the Solution (continued)

Theorem 1 (Gauss-Markov)

Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ be a multiset of examples to be fitted with a linear model function as $y(\mathbf{x}) \overset{(\star)}{=} \mathbf{x}^T\mathbf{w}$. Within the class of linear unbiased estimators for $\mathbf{w}$, the least squares estimator $\hat{\mathbf{w}}$ has minimum variance, i.e., is most efficient.

Related follow-up issues:

- mean and variance of $\hat{\mathbf{w}}$
- proof of the Gauss-Markov theorem
- weak set and strong set of assumptions
- efficiency and consistency of unbiased estimators
- rank deficiencies, where the feature number p exceeds $|D| = n$
- relation between least squares and maximum likelihood estimators / methods

Remarks:

- The Gauss-Markov Theorem is important since it follows already from the weak set of assumptions.
- Under the strong set of assumptions the maximum likelihood estimates are identical to the least-squares estimates.