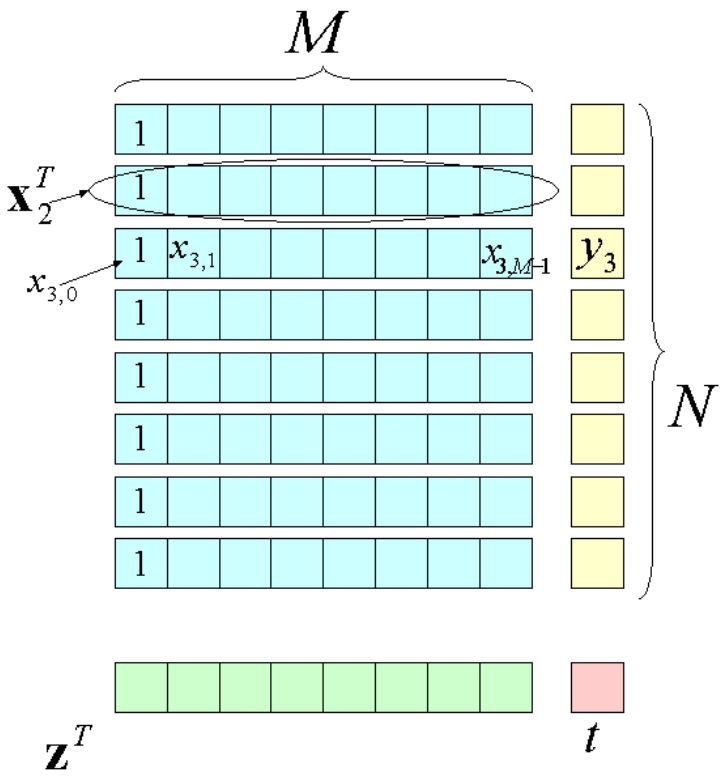
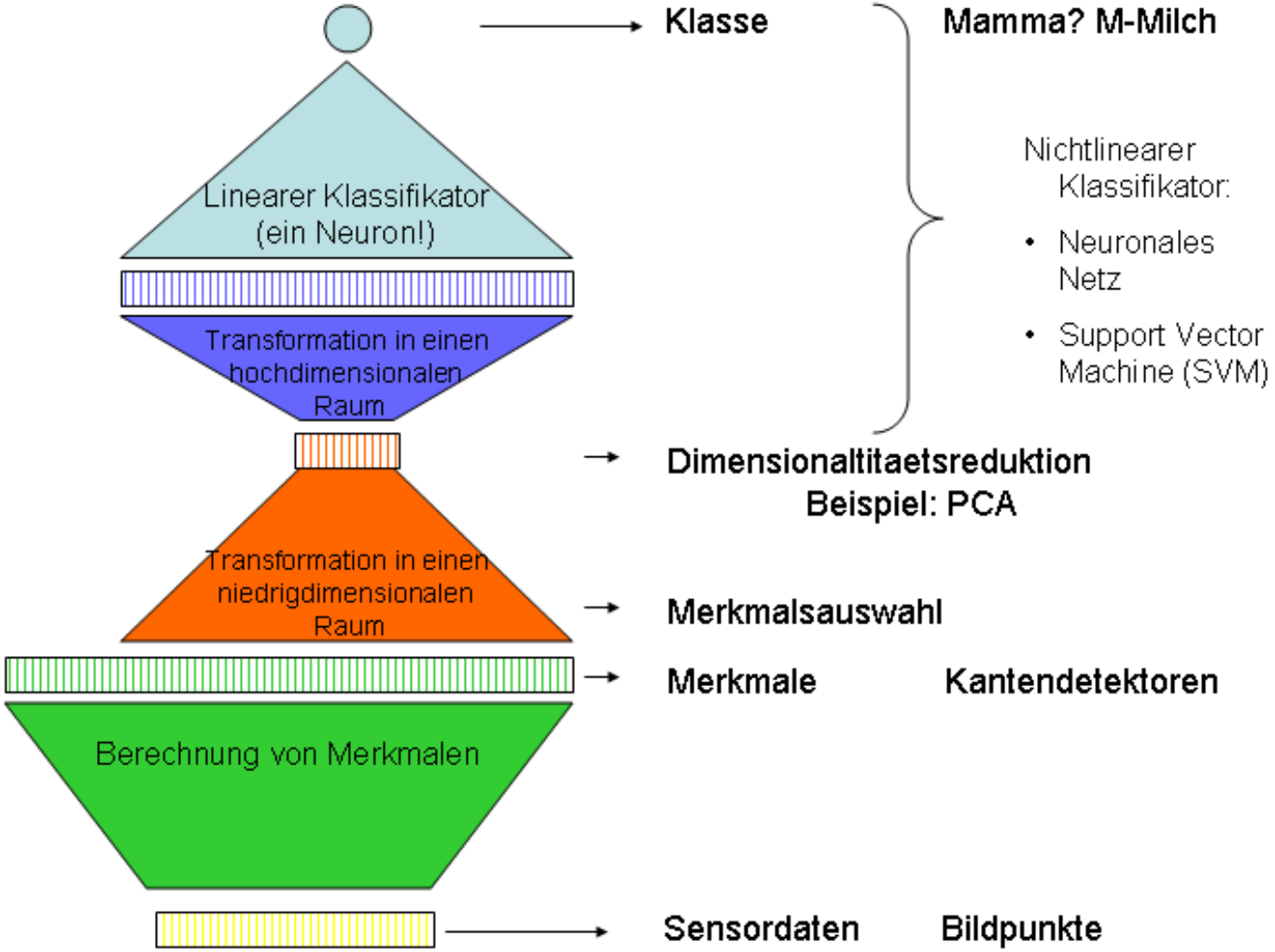
Die Datenmatrix für Überwachtes Lernen



X_j	j-te Eingangsvariable
X	$= (X_0, \ldots, X_{M-1})^T$ Vektor von Eingangsvariablen
M	Anzahl der Eingangsvariablen
N	Anzahl der Datenpunkte
Y	Ausgangsvariable
$\mathbf{x}_i$	$= (x_{i,0}, \ldots, x_{i,M-1})^T$ i-ter Eingangsvektor
$x_{i,j}$	j-te Komponente von $\mathbf{x}_i$
y_i	i-te Zielgröße
$\mathbf{d}_i$	$= (x_{i,0}, \ldots, x_{i,M-1}, y_i)^T$ i-tes Muster
D	$= \{\mathbf{d}_1, \ldots, \mathbf{d}_N\}$ (Trainings-) Datensatz
$\mathbf{z}$	Testeingangsvektor
t	Unbekannte Testzielgröße zu $\mathbf{z}$
$\mathbf{X}$	$= (\mathbf{x}_1, \ldots \mathbf{x}_N)^T$ design matrix

Das Lernmodell



Lineare Regression

Volker Tresp

Verschiedene statistische Prinzipien des Maschinellen Lernens

- Minimierung des empirischen Risikos
  - Grundlage der *Statistischen Lerntheorie*
  - Regression: Methode der kleinsten Quadrate (Gauss)
- Regularisierungstheorie

Empirische Risiko Minimierung (ERM)

- Überwachtes Lernen: Die Zielgröße Y soll anhand von Eingangsvariablen $\mathbf{X}$ vorhergesagt werden
- Die einzige wesentliche Annahme ist, dass $P(\mathbf{x}, y)$ stationär (fest und unbekannt) ist
- Man definiert eine Klasse von Lernmaschinen (Funktionenklasse)
  - Beispiel: Funktionen $f(\mathbf{x}, \mathbf{w})$ mit Parametervektor $\mathbf{w}$
- Man definiert eine Verlustfunktion (Fehlerfunktion). Bei der Regression ist der quadratische Fehler gebräuchlich

$$\text{loss}(y, f(\mathbf{x}, \mathbf{w})) = (y - f(\mathbf{x}, \mathbf{w}))^2$$

Empirische Risiko Minimierung (2)

- Ziel ist es, aus der Menge der ausgewählten Funktionenklasse diejenige Funktion zu finden, die den erwartete Verlust minimiert, der durch das Risikofunktional

$$R(\mathbf{w}) = \int \text{loss}(y, f(\mathbf{x}, \mathbf{w}))P(\mathbf{x}, y)\ d\mathbf{x}dy$$

  definiert ist

- Im Falle des quadratischen Fehlermaßes ergibt sich

$$R(\mathbf{w}) = \int (y - f(\mathbf{x}, \mathbf{w}))^2 P(\mathbf{x}, y)\ d\mathbf{x}dy$$

Empirische Risiko Minimierung (3)

- In der Wahrscheinlichleitslehre (Probability) nimmt man an, dass $P(\mathbf{x}, y)$ bekannt ist; eine typische Aufgabe ist es dann, z.B. den besten linearen Schätzer zu finden
- In der Statistik ist $P(\mathbf{x}, y)$ unbekannt; man kennt nur einen Trainingsdatensatz (Stichprobe, *sample*) der Größe N,

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$$

- Wir nehmen an, dass die Daten i.i.d. (independent, identically distributed) sind

Empirische Risiko Minimierung (4)

- Folgt man dem Prinzip der empirischen Risiko Minimierung (empirical risk minimization), minimiert man im Training das empirische Risiko

$$R(\mathbf{w}) \approx R_{emp}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \text{loss}(y, f(\mathbf{x}, \mathbf{w}))$$

- Definiert man als Verlustfunktion den quadratischen Fehler, ergibt sich als empirisches Risiko der mittlere quadratische Fehler der Trainingsdaten

$$R_{emp}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - f(x_i, \mathbf{w}))^2$$

wobei wir auch gleich definieren

$$J_N(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f(x_i, \mathbf{w}))^2 = N \times R_{emp}(\mathbf{w})$$

Empirische Risiko Minimierung und das Prinzip der kleinsten Quadrate

- Wählt man als Verlustfunktion den quadratischen Abstand, so reduziert sich das ERM Prinzip reduziert auf die Methode der kleinsten Quadrate, *least squares (LS) principle*)

Kleinste-Quadrate Schätzer für lineare Regression (eindimensional)

Eindimensionales Modell:

$$f(x, \mathbf{w}) = w_0 + w_1 x$$

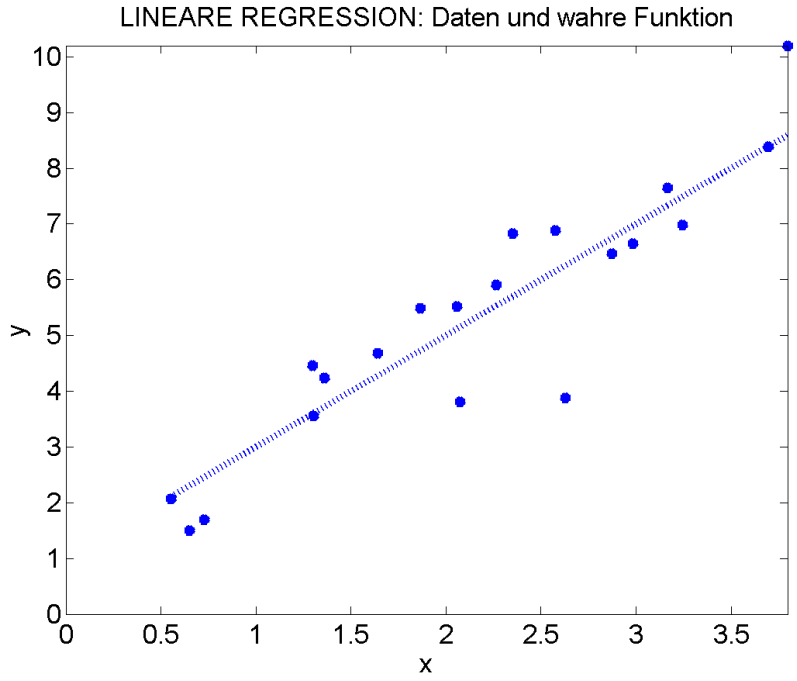
$$\mathbf{w} = (w_0, w_1)^T$$

Empirischer quadratischer Fehler:

$$J_N(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f(x_i, \mathbf{w}))^2$$

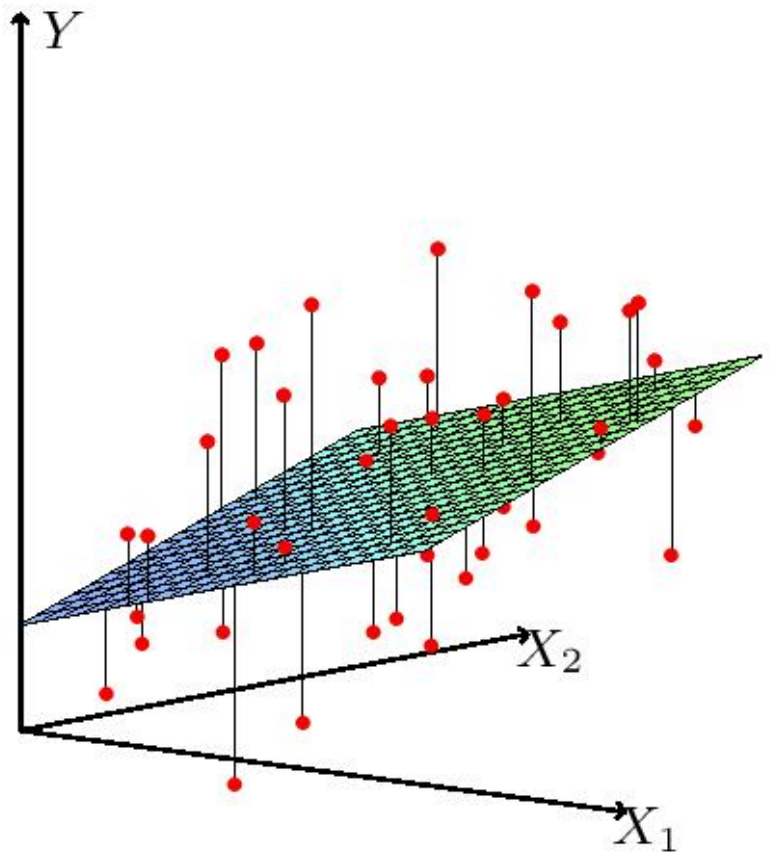
Finde:

$$\mathbf{w}_{LS} = \arg\min_{w} J_N(\mathbf{w})$$

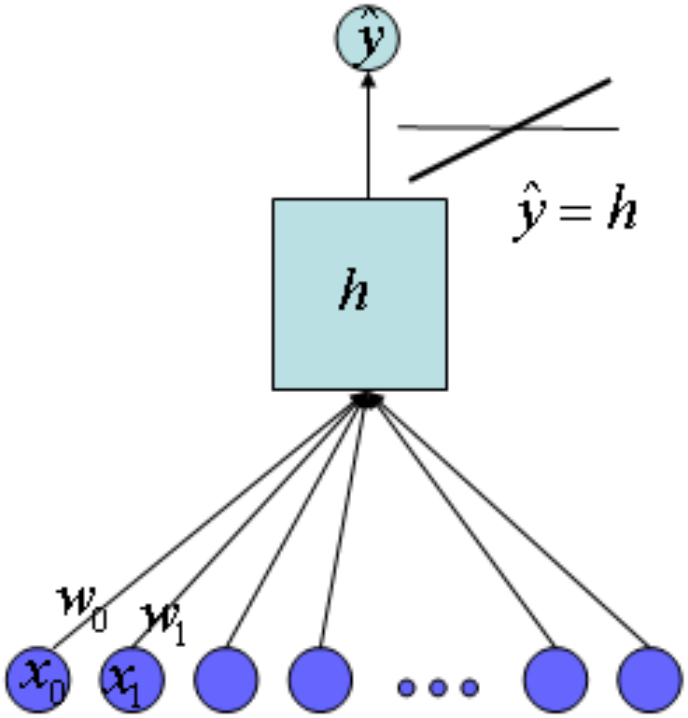


$$w_0 = 1, w_1 = 2, var(\epsilon) = 1$$

Mehrdimensionale Lineare Regression



Das Lineare Neuron



- Ein lineares Modell kann man als Neuron mit linearer Übertragungsfunktion interpretieren (Adaline)
- Zunächst wird die Aktivierungsfunktion als gewichtete Summe der Eingangsgrössen x_i berechnet zu

$$h = \sum_{j=0}^{M-1} w_j x_j$$

- Das lineare Neuron unterscheidet sich vom Perceptron durch die Übertagungsfunktion

$$\textbf{Perceptron}: \quad \hat{y} = \textbf{sign}(h)$$

$$\textbf{LinearesNeuron}: \quad \hat{y} = h$$

Kleinste-Quadrate Schätzer für Regression (mehrdimensional)

Mehrdimensionales Modell:

$$f(\mathbf{x}_i, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j x_{i,j}$$

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

$$\mathbf{w} = (w_0, w_1, \ldots w_{M-1})^T$$

$$\mathbf{x}_i = (1, x_{i,1}, \ldots, x_{i,M-1})^T$$

LS-Lösung

Empirischer quadratischer Fehler:

$$J_N(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i, \mathbf{w}))^2$$

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

$$\mathbf{y} = (y_1, \ldots, y_N)^T$$

$$\mathbf{X} = \begin{pmatrix} x_{1,0} & \cdots & x_{1,M-1} \\ \cdots & \cdots & \cdots \\ x_{N,0} & \cdots & x_{N,M-1} \end{pmatrix}$$

LS-Lösung (2)

Matrix calculus:

$\mathbf{y}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$
$\mathbf{Ax}$	$\mathbf{A}^T$
$\mathbf{x}^T\mathbf{A}$	$\mathbf{A}$
$\mathbf{x}^T\mathbf{x}$	$2\mathbf{x}$
$\mathbf{x}^T\mathbf{Ax}$	$\mathbf{Ax} + \mathbf{A}^T\mathbf{x}$

Daher

$$\frac{\partial J_N(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial(\mathbf{y} - \mathbf{X}\mathbf{w})}{\partial w} \times 2(\mathbf{y} - \mathbf{X}\mathbf{w}) = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})$$

LS-Lösung (3)

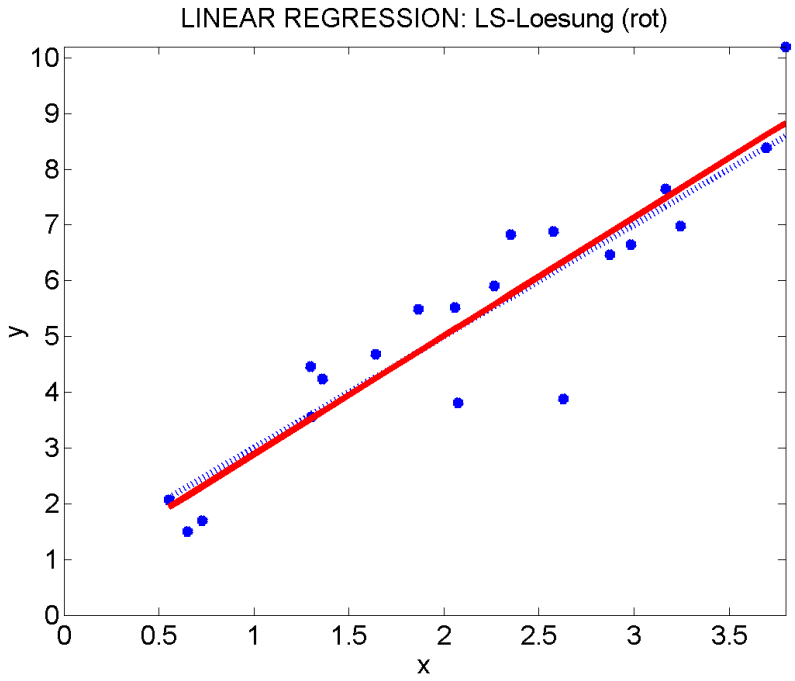
Berechnung der LS-Lösung:

$$\frac{\partial J_N(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$$

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

Komplexität (linear in N):

$$\mathcal{O}(M^3 + NM^2)$$



$$\hat{w}_0 = 0.75, \hat{w}_1 = 2.13$$

Diskussion: Empirische Risiko Minimierung

- Der Vorteil des ERM Prinzips ist, dass keine Annahmen über das zugrundeliegende "wahre" datengenerierende Model gemacht werden müssen
- Die einzige wesentliche Annahme ist, dass $P(\mathbf{x}, y)$ stationär ist
- Nachteil: für ein endliches N wird ein zu komplexes Modell ausgewählt (Überanpassung, overfitting)
- Dies zeigt sich ebenso daran, dass $\hat{\mathbf{w}}_{LS}$ sehr instabil sein kann (wenn $M \approx N$), das heißt, sehr empfindlich auf kleine Änderungen der Daten reagiert
- Diesem Problem behilft man sich durch Einführung eines Strafterms

$$R_{emp}(\mathbf{w}) + \text{complexity term}$$

- Regularisierungstheorie: theory of ill-conditioned problems
- Beispiel: $\text{complexity term} = \lambda \mathbf{w}^{\mathbf{T}} \mathbf{w} = \lambda \sum_i w_i^2$, mit $\lambda \geq 0$

- Das ERM Prinzip ist die Grundlage für die *Statistical Learning Theory* (VC-Theory)
- Hier zeigt man, dass mit hoher Wahrscheinlichkeit

$$R(\mathbf{w}) \leq R_{emp}(\mathbf{w}) + \text{complexity term}(\text{VC-dimension})$$

  und man wählt die Funktion aus, die rechte Seite minimiert (mehr später)

Regularisierungstheorie

- Inverse Probleme sind häufig schlecht gestellt (ill-posed): hier: Die Lösung hängt nicht stetig von den Daten ab
- Um das Problem numerisch zu lösen führt man Zusatzannahmen ein: Glattheit, ...
- Tikhonov Regularisierung (Andrey Nikolayevich Tychonoff): Kompromiss zwischen Anpassung an die Daten und einer Reduktion der Norm: Ridge Regression
- Regularisierungstheorie: minimale Annahmen; ohne Bezug zu Probability oder Statistik!
- Bayes'sche Interpretation: Stetigkeitsannahmen als a prior Annahme

Lineare Regression und Regularisierung

- Regularisierte Kostenfunktion (penalized least squares (PLS), Ridge Regression, Weight Decay): der Einfluss einer Eingangsgröße sollte klein sein

$$J_N^{pen}(\mathbf{w}) = \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i, \mathbf{w}))^2 + \lambda \sum_{i=0}^{M-1} w_i^2$$

$$\widehat{\mathbf{w}}_{Pen} = \left(\mathbf{X}^T\mathbf{X} + \lambda I\right)^{-1}\mathbf{X}^T\mathbf{y}$$

Herleitung:

$$\frac{\partial J_N^{pen}(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + 2\lambda\mathbf{w} = 2[-\mathbf{X}^T\mathbf{y} + (\mathbf{X}^T\mathbf{X} + \lambda I)\mathbf{w}]$$

Lineare Regression: Regularisierung

- Regularisierung verringert den Einfluss von Kollinearität
- Kollinearität erhöht die Anzahl der Freiheitsgrade ohne neue Information einzubringen
- Die zusätzlichen Freiheitsgrade werden dazu verwendet, um "das Rauschen" zu fitten
- Regularisierung beschränkt die Freiheitsgrade sinnvoll

Beispiel

- Drei Datenpunkte werden generiert nach

$$y = 0.5 + x_1 + \epsilon$$

- (korrekten) Modell 1

$$y = w_0 + w_1 x_1 + \epsilon$$

- Korrelierter weiterer Eingang

$$x_2 = x_1 + \delta$$

- Modell 2

$$y = w_0 + w_1 x_1 + w_2 x_2 + \epsilon$$

Beispiel (2)

Daten, die Modell 1 sieht:

x_1	y
-0.2	0.49
0.2	0.64
1	1.39

Daten, die Modell 2 sieht:

x_1	x_2	y
-0.2	-0.1996	0.49
0.2	0.1993	0.64
1	1.0017	1.39

Beispiel (3)

Gewichte:

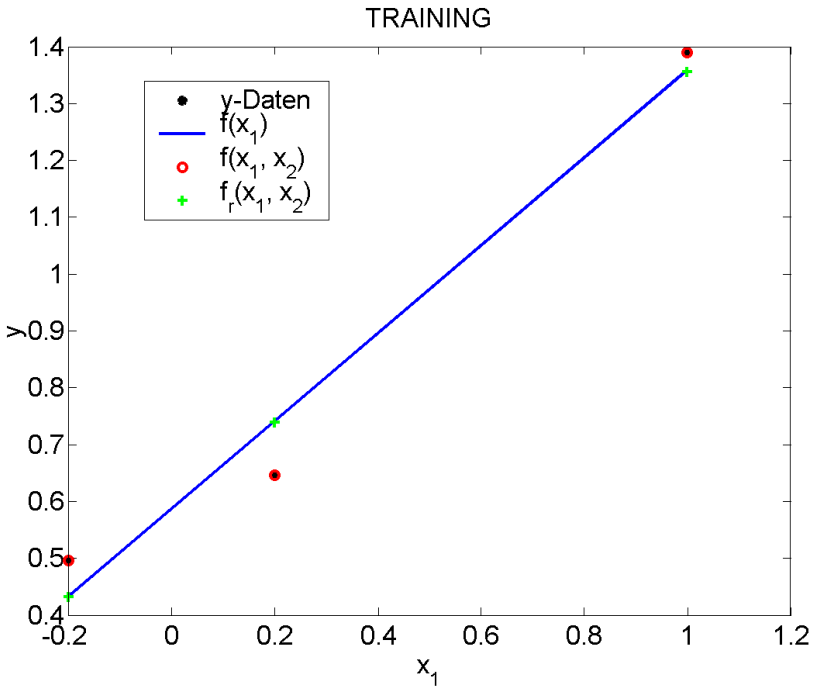
w	$\widehat{w}_{ML}$	$\widehat{w}_{ML,2}$	$\widehat{w}_{pen,2}$
0.5	0.58	0.67	0.58
1	0.77	-136	0.38
		137	0.39

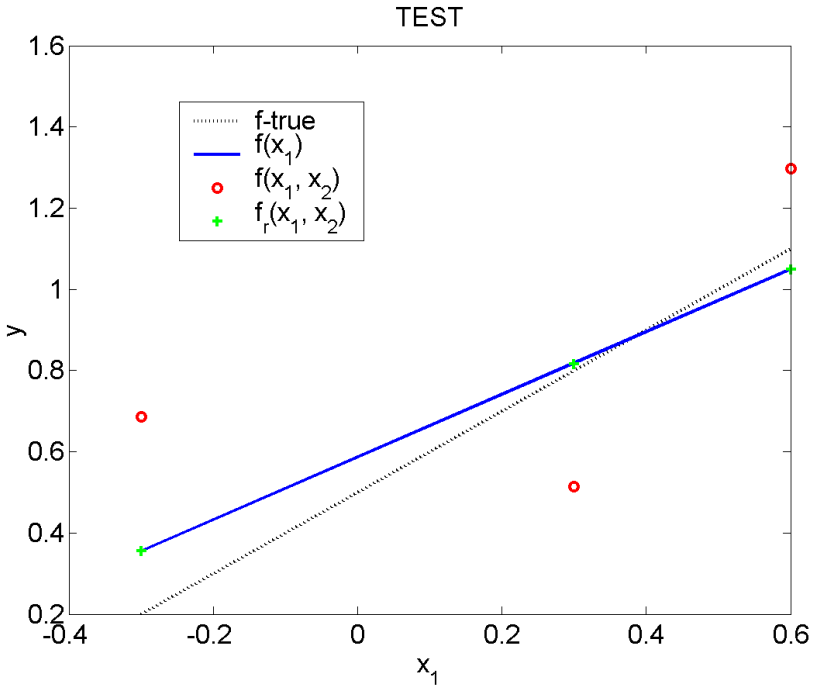
Training:

y	f_{ML}	$f_{ML,2}$	$f_{pen,2}$
0.50	0.43	0.50	0.43
0.65	0.74	0.65	0.74
1.39	1.36	1.39	1.36

Test:

y_{true}	f_{ML}	$f_{ML,2}$	$f_{pen,2}$
0.20	0.36	0.69	0.36
0.80	0.82	0.51	0.82
1.10	1.05	1.30	1.05





Prostate Cancer Data

8 Inputs, 97 Data Points; y: prostate-specific antigen; $M_{eff} = 4.16$

10-fach Kreuzvalidierung	LS	0.586
	Best Subset (3)	0.574
	Ridge (Weight Decay)	0.540