Einführung in die Wissensverarbeitung

2 VO 708.560 + 1 UE 442.072

SS 2013

Institut für Signalverarbeitung und Sprachkommunikation

TU Graz
Inffeldgasse 12/1
www.spsc.tugraz.at

Institut für Grundlagen der Informationsverarbeitung

TU Graz
Inffeldgasse 16b/1
www.igi.tugraz.at

Organisation

Vortragende:

Vorlesung:

Dr. Franz Pernkopf (SPSC)

Dr. Stefan Häusler (IGI)

Übungen:

DI Paul Meissner(SPSC)

DI Stefan Habenschuss (IGI)

Allgemein:

VO: Mittwoch, 14:15, HSi13

UE: Dienstag, 13:15 – 18:00 (3 Gruppen), HSi11

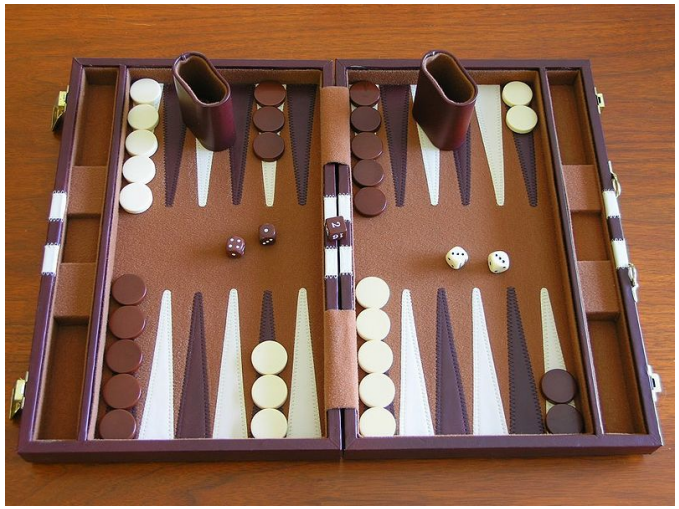
Webpage www.spsc.tugraz.at/courses/ew

Newsgroup: tu-graz.lv.ew

Maschinelles Lernen

Arthur Samuel (1959):

„Machine learning: Field of study that gives computers the ability to learn without being programmed.“



Problem:
Auf Grund der Komplexität des Spiels ist die optimale Strategie nicht bekannt.

Was ist Wissensverarbeitung?

„Künstliche“ Generierung von Wissen aus Erfahrung:

- Entdecken und Strukturieren von Wissen,
- Ableiten von neuem Wissen,
- Kommunikation dieses Wissens.
- Repräsentation dieses Wissens (im Computer)

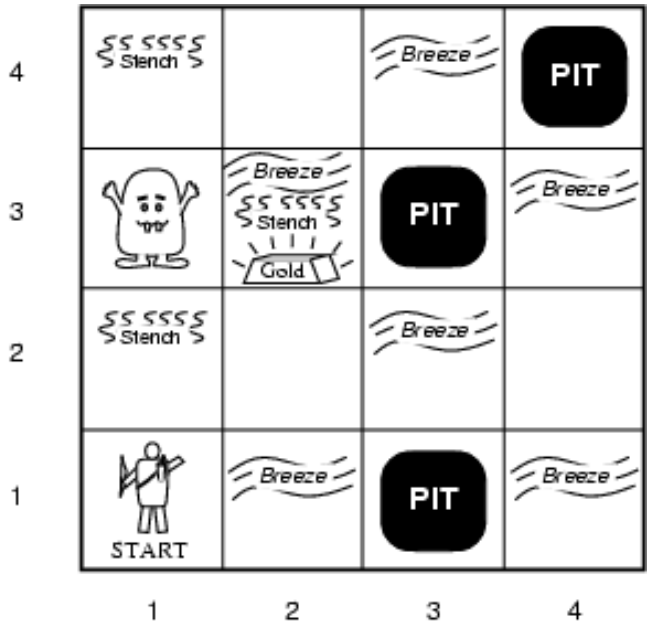
Formen der Wissensverarbeitung

- Wissensverarbeitung mit sprachlichen Strukturen.
  - **Logische Systeme**:
    Aussagenlogik und Prädikatenlogik

- Heuristische Ansätze, wie z.B. aus den Gebieten Mustererkennung, Neuronale Netze, evolutionäre Algorithmen etc.
  - **Computational Intelligence:**
    Formulierung von präzisen logischen Aussagen ist praktisch nicht möglich. Anwendung von heuristischen Methoden

Wissensverarbeitung mit logischen Systemen

- Entwurf von Agenten, welche die **Welt repräsentieren** können
- ... und eines **Schlußfolgerungsprozesses**, welcher neue Repräsentationen ableitet und diese dazu nutzt, um zu schließen, was zu tun ist.



Wumpus-Welt

Computational Intelligence

CI wird angewendet, wenn die Lösung einer Aufgaben schwer mit logischen Sprachen oder klaren Regeln formuliert werden kann.

Beispiel: Bilderkennung

Meissner

Pernkopf

Neumann

Häusler

Anwendungsbereiche

Recommender systems

- Amazon.com

- Online Radio www.last.fm
- Netflix US online Filmverleih (Datensatz mit 100 Mio. Einträge)

Anwendungsbereiche

NETFLIX

Netflix Prize, 1 Mio. US-Dollar für 10% Vorhersagevebesserung

Weitere Anwendungsbereiche ...

- Gesichts-, Sprach-, Schrifterkennung (PLANET Addresserkennung)
- Spam-Erkennung
- Notierung biologischer Sequenzen, Moleküle, chem. Proben (Innocentive)
- Marktanalysen (e.g. Aktienkursvorhersagen)
- Fahrzeug- und Personenzählsysteme (Austrian Research center)
- Fahrzeugsteuerungen (autonom. Fahrzeug von Google, Zulassung in US)
- u.s.w.

Lehrveranstaltungsübersicht

IGI

Kapitel 1	Grundbegriffe des maschinellen Lernens
Kapitel 2	Neuronale Netze
Kapitel 3	Klassische Klassifikationsalgorithmen
Kapitel 4	Modellselektion
Kapitel 5	Aussagenlogik

SPSC

Unüberwachtes Lernen, *Hidden Markov Models*, Autom. Spracherkennung

Plan für heute: Kapitel 1

- Welche Arten des Lernens gibt es?
- Wie kann man überwachtes Lernen formalisieren?
- Was genau ist das Lernziel?
- Wie werden überwachte Lernprobleme allgemein gelöst (Ablauf)?
- Lernalgorithmus „lineare Regression“

Welche Arten des Lernens gibt es?

- **Supervised learning (überwachtes Lernen):**

  Gegeben: Trainingsbeispiele mit Zielwerten (z.B. Zuordnungen)

  Ziel: Zielwertvorhersage für neue Beispiele (z.B. Marktpreise)

- **Unsupervised learning (unüberwachtes Lernen):**

  Gegeben: Trainingsbeispiele ohne Zielwerte

  Ziel: Erkennen von Struktur in den Daten

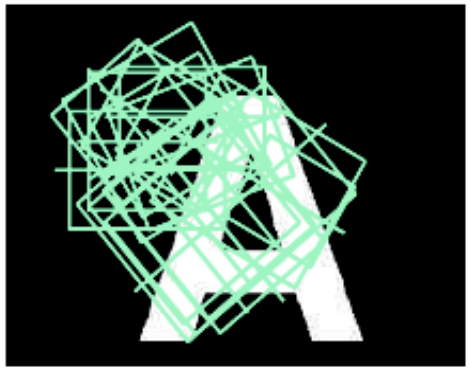
- **Reinforcement learning (verstärkendes Lernen):**

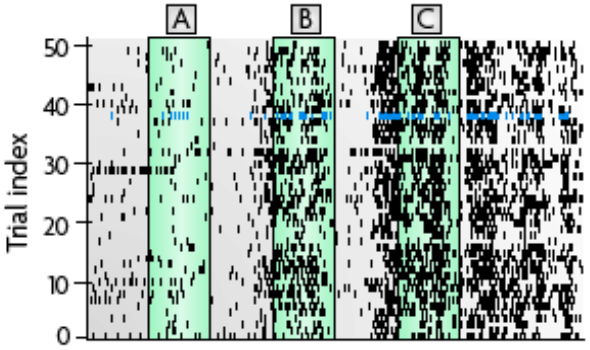
  Gegeben: Trainingsbeispiele mit feedback für gewählte Handlungen

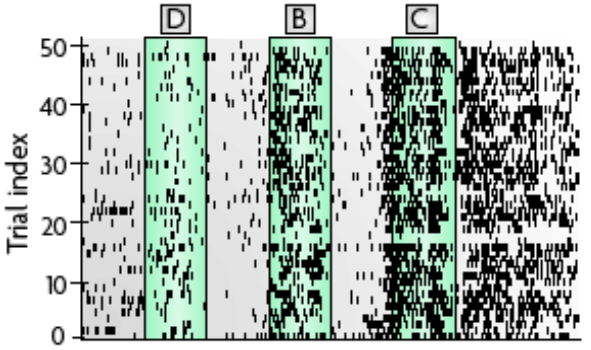
  Ziel: Minimierung der Kosten von Handlungssequenzen

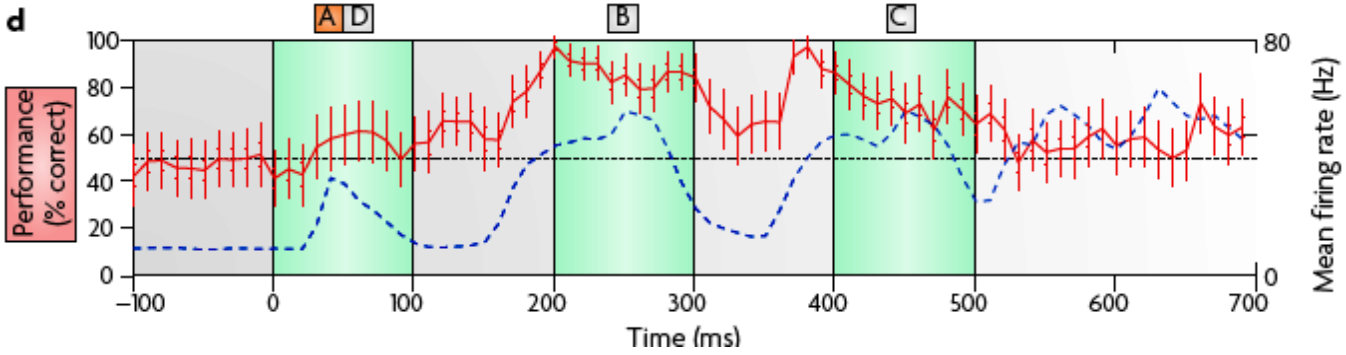
Beispiel überwachtes Lernen

Neurowissenschaft: Klassifikation von visuellen Stimuli



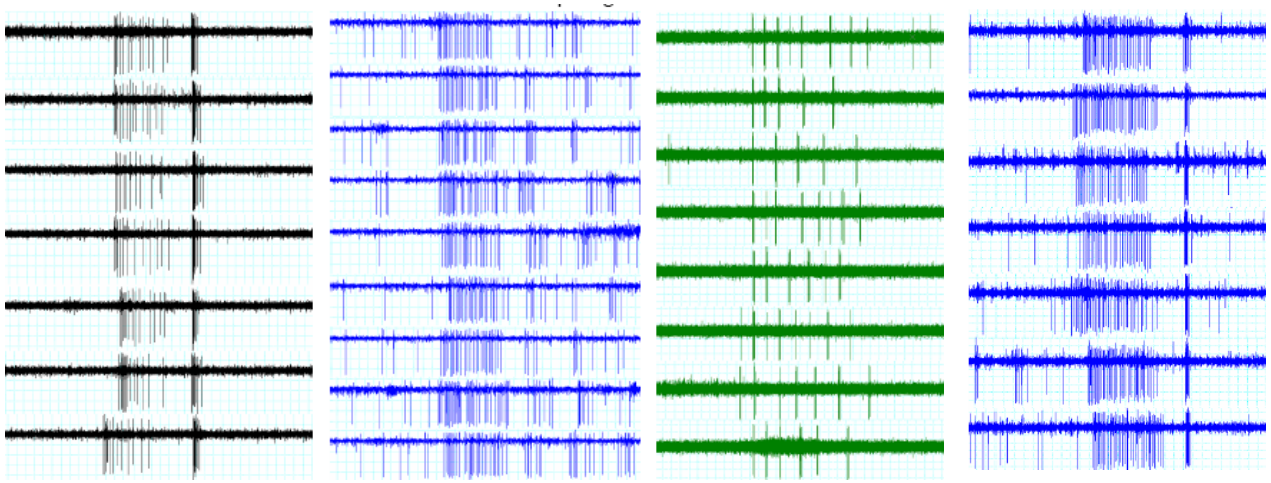






Beispiel unüberwachtes Lernen

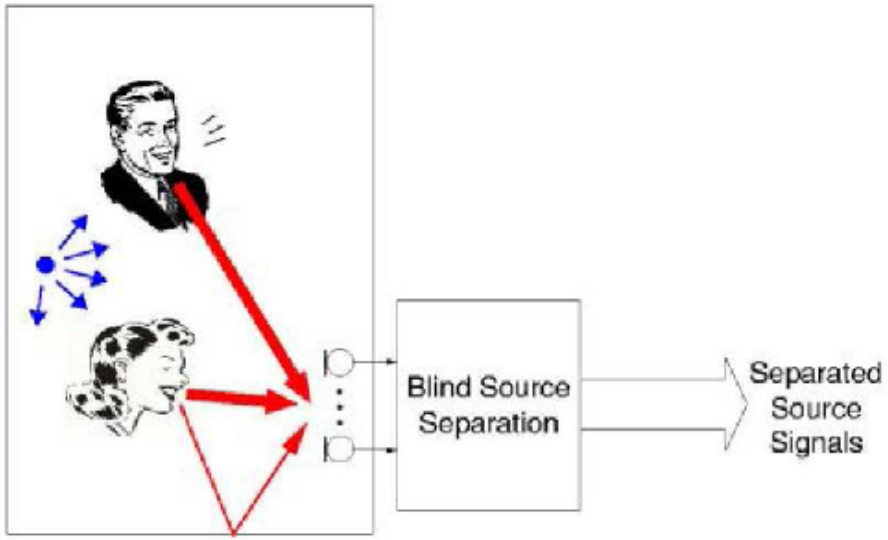
Cluster-Analyse

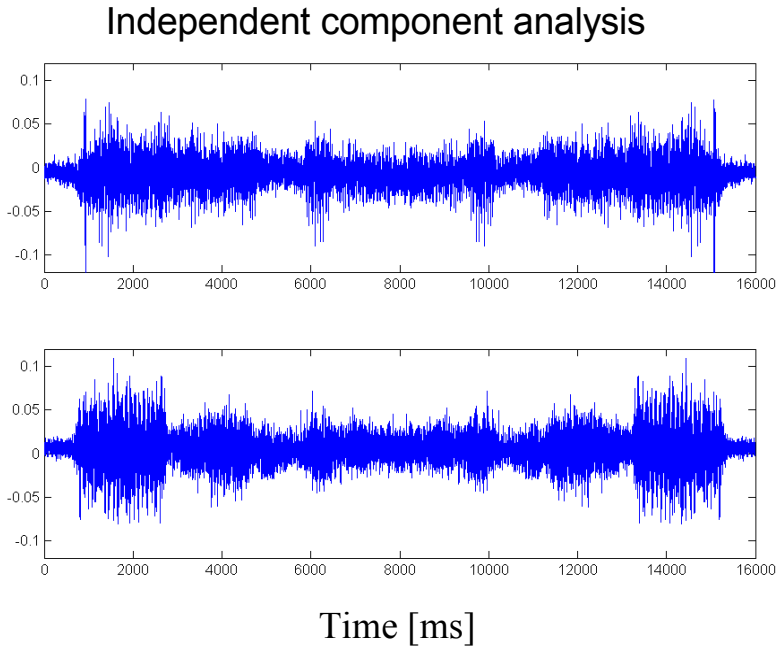


Daten: Entladungsmuster eines auditorischen Neurons der Laubheuschrecke
in Reaktion auf verschiedene Schallquellen (Artgenossen, Feinde, ...)

Beispiel unüberwachtes Lernen

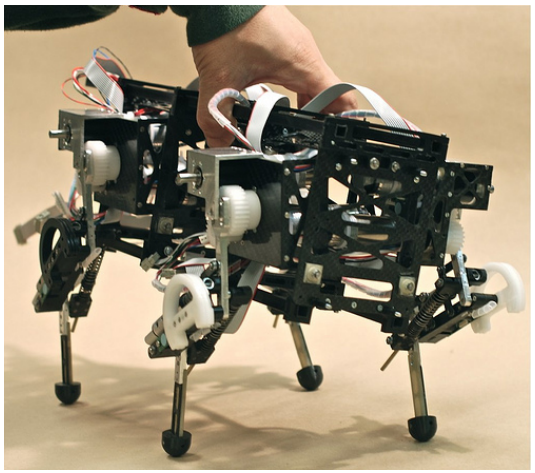
Blind source separation (Cocktailparty Problem)





Beispiel verstärkendes Lernen

Steuerung von autonomen Helikoptern.



(EU Projekt Amarsi)

Kapitel 1

- Wie kann man überwachtes Lernen formalisieren?

Beispiel Gesichtserkennung

Gegeben: Kollektion von Bildern und Kennzeichnungen.

Meissner, +

Pernkopf, +

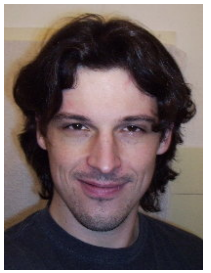
Neumann, +

Häusler, -

Kennzeichnungen (labels):

+ … Gebäudezutritt
- … Kein Gebäudezutritt

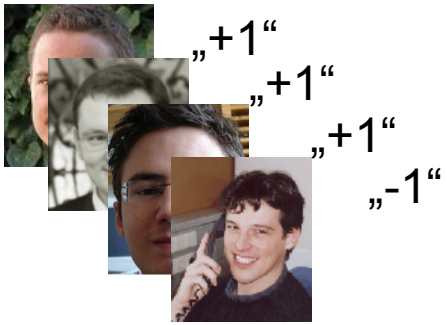
Ziel: Korrektes Klassifizieren neuer Bilder.



Häusler, - ?

Lernbeispiele f. überwachtes Lernen

Gegeben: Eingabewerte und Ausgabewerte.



$$X = \mathbb{R}^d, \quad Y = \{-1, +1\}$$

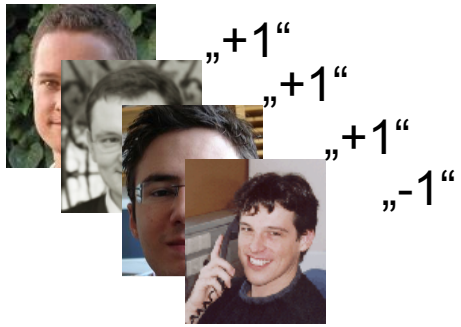
Formalisierungsschritte

- Hypothese
- Lernalgorithmus
- Fehlerkriterium

Hypothese

Hypothese, *die; -, -n [hü ..] (griech.) noch unbewiesene, als Hilfsmittel für eine Erkenntnis benutzte Annahme, Vermutung.*

Eine Hypothese H führt Vorhersagen durch.



→ „+1“

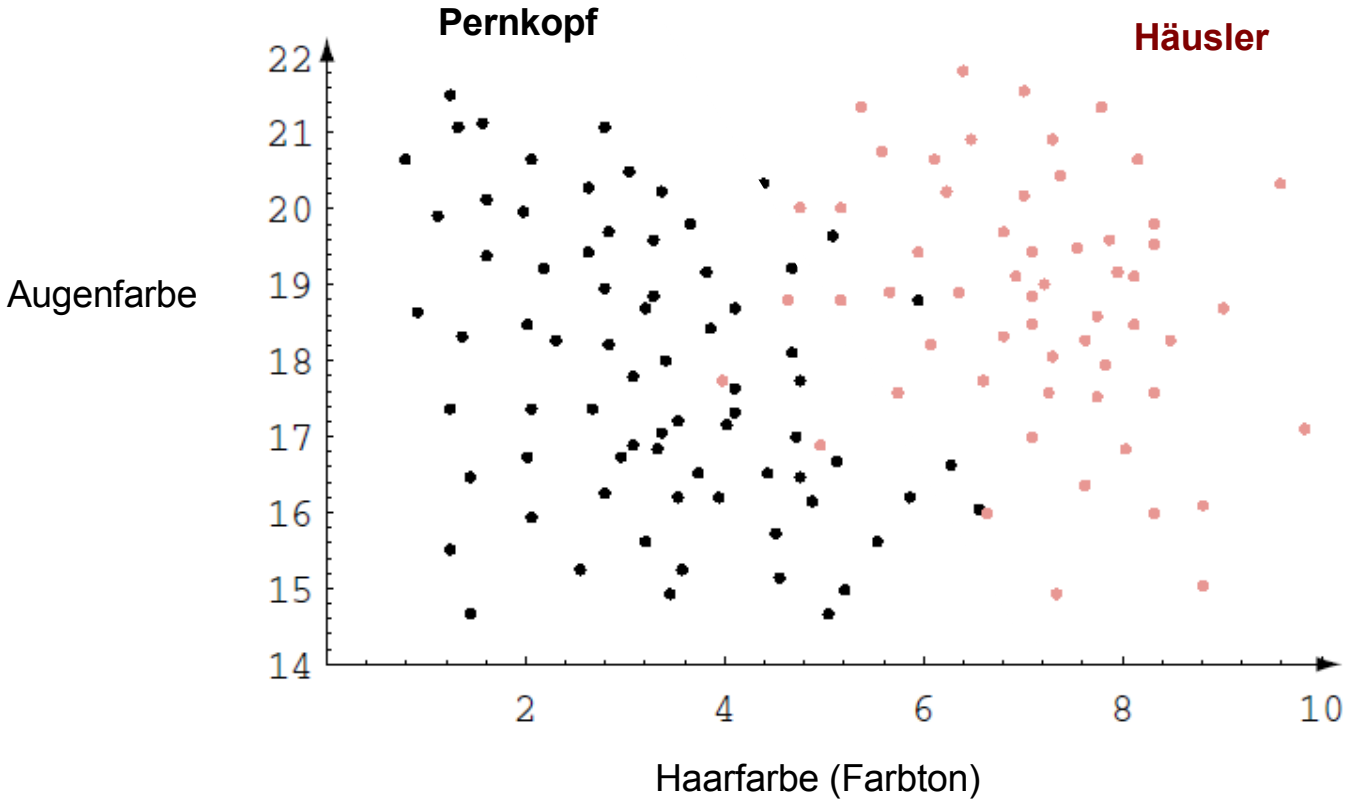
$$X = \mathbb{R}^d, \quad Y = \{-1, +1\}$$

Klassifikation

Klassifikation ist die Zuweisung von Eingabewerten zu diskreten Ausgabewerten:

$$H : X \rightarrow Y \qquad \text{wobei } |Y| \text{ endlich ist.}$$

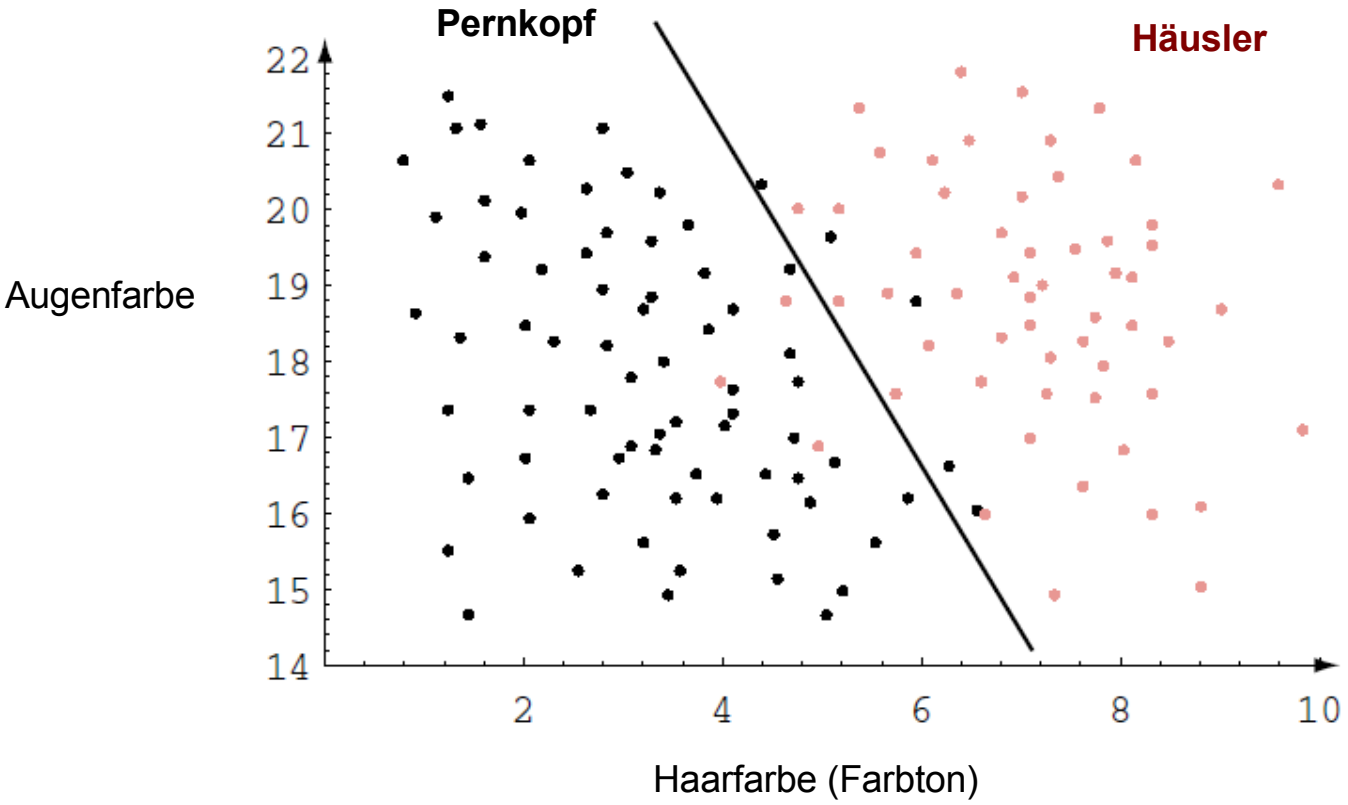
Beispiel:



Lineare Klassifikation

Einteilungen von Eingabewerten in Klassen basierend auf einer linearen Kombination der Eingabewerte.

Beispiel: $\mathbb{R}^2$

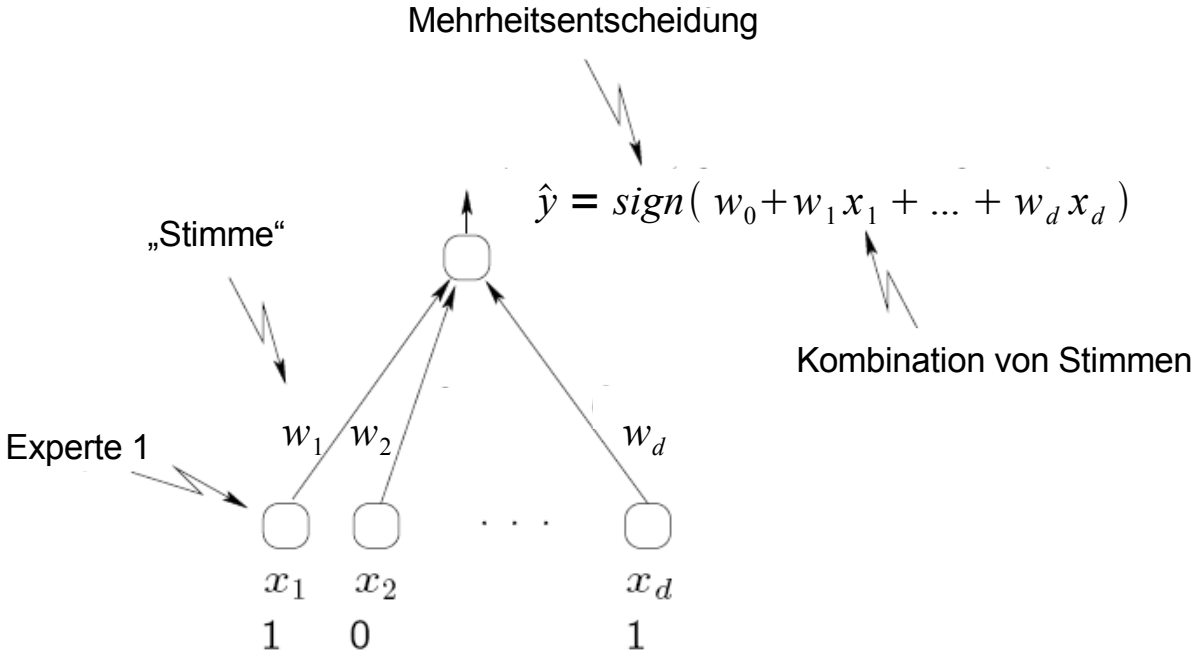


Binäre lineare Klassifikation

Einteilungen von Eingabewerten in 2 Klassen anhand des linearen Klassifikators

$$\hat{y} = sign(\boldsymbol{w}^T \boldsymbol{x}) = sign(w_0 + w_1 x_1 + ... + w_d x_d) \qquad \boldsymbol{x}, \boldsymbol{w} \in \mathbb{R}^d$$

Interpretation:
Gewichtete Kombination von Expertenmeinungen (z.B. für binäre Attribute)



Hypothesenklassen

Eine Hypothesenklasse $\mathcal{H}$ ist eine Menge von Hypothesen.

Z.B: Linearer Klassifikator

Hypothesenklasse wird mit dem Vektor $\boldsymbol{w} \in \mathbb{R}^d$ parametrisiert.

$$\hat{y} = sign(\boldsymbol{w}^T \boldsymbol{x}) = sign(w_0 + w_1 x_1 + ... + w_d x_d)$$

Lernalgorithmus

Ein Lernalgorithmus $\mathcal{A}$ wählt eine Hypothese H aus der Hypothesenklasse $\mathcal{H}$ anhand einer Liste L von l Trainingsbeispielen aus.

Gegeben: $L = ((\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l))$

$$L \in (X \times Y)^*$$

$$(X \times Y)^* := \{(s_1, \ldots, s_l) \mid l \geq 0 \text{ und jedes } s_i \in (X \times Y)\}$$

Fehlerkriterium

Das Fehlerkriterium $E(y, \hat{y})$ quantifiziert die Abweichung der Ausgabe der Hypothese $\hat{y}$ vom Sollwert y .

Kapitel 1

- Was genau ist das Lernziel?

Empirischer Fehler

Der empirische Fehler ist der Mittelwert des Fehlerkriteriums aller Lernbeispiele

Naheliegend:

Finde w welches

Empirischer Fehler

$$error_L = \overbrace{\frac{1}{l} \sum_{k=1}^{l} E(y_k, H(\boldsymbol{w}, \boldsymbol{x}_k))}$$

minimiert.

Warum ist es sinnvoll den empirischen Fehler zu minimieren?

Wahrer Fehler

Der empirische Fehler dient zur Abschätzung des wahren Fehlers.

Empirischer Fehler: $error_L = \frac{1}{l} \sum_{k=1}^{l} E(y_k, H(\boldsymbol{w}, \boldsymbol{x}_k))$

Wahrer Fehler: $?$

In welcher Beziehung stehen der empirische und der wahre Fehler?

Analyse des wahren Fehlers

Für eine beliebige Hypothese H und gezogenes Paar $(\boldsymbol{x}, y)$ kann man die Wahrscheinlichkeit eines Fehlers $p := error_P(H)$ mit einem Münzwurfexperiment vergleichen („Bernoulli-Experiment“).

$$H(\boldsymbol{x}) = y \rightarrow E = 0$$

$$H(\boldsymbol{x}) \neq y \rightarrow E = 1$$

Analyse des wahren Fehlers

Der empirische Fehler für n Testbeispiele T_n entspricht n Münzwürfen

(identisch verteilte und unabhängige Zufallsvariablen)

$$error_{T_n}(H) = \frac{S}{n} \qquad S = \sum_{k=1}^{n} E_k$$

Lernphasen

Trainingsphase: Dient zur Auswahl der Hypothese aus einer Hypothesenklasse $\mathcal{H}$ durch einen **Lernalgorithmus** anhand von Trainingsdaten.

Testphase: Dient zur Betimmung der Qualität der ausgewählten Hypothese durch ein **Fehlerkriterium** anhand von Testdaten (Generalisierung).

$$\text{Trainingsdaten} \cap \text{Testdaten} = \emptyset$$

Anforderungen für gute Vorhersagen

Welche Anforderungen stellen wir an Hypothesen und Lernalgorithmen in Anbetracht dieser Überlegungen.

- Was ist eine gute Hypothese?
- Was ist ein guter Lernalgorithmus?

Qualität einer Hypothese

Die Qualität einer Hypothese wird anhand des wahren Fehlers gemessen.

$$error_P = \langle E(y, H(\boldsymbol{w}, \boldsymbol{x})) \rangle_{(\boldsymbol{x}, y) \sim P}$$

Dieser wird anhand unabhängiger Testdaten T abgeschätzt, welche nicht zum Trainieren (d.h. zur Auswahl der Hypothesenklasse) verwendet wurden.

$$error_{T_n} = \frac{1}{n} \sum_{k=1}^{n} E(y_k, H(\boldsymbol{w}, \boldsymbol{x}_k))$$

Qualität eines Lernalgorithmus

- Die Qualität eines Lernalgorithmus $\mathcal{A}$ wird zum einen daran gemessen, ob es in der Hypothesenklasse $\mathcal{H}$ überhaupt eine Hypothese H mit niedrigem wahren Fehler $error_P(H)$ gibt („expressibility of $\mathcal{H}$“).

- Zum anderen wird die Qualität daran gemessen, wie groß die Chance ist dass $\mathcal{A}$ schon für eine relativ kurze Liste L von Trainingsbeispielen eine Hypothese H in $\mathcal{H}$ findet, deren wahrer Fehler $error_P(H)$ nicht viel größer als der empirische Fehler $error_L(H)$ ist.

Kapitel 1

- Fallbeispiel: Lineare Regression.

Regression

Das Ziel ist Zuweisung (quantitative Vorhersage) von Eingabewerten zu reellwertigen Ausgabewerten:

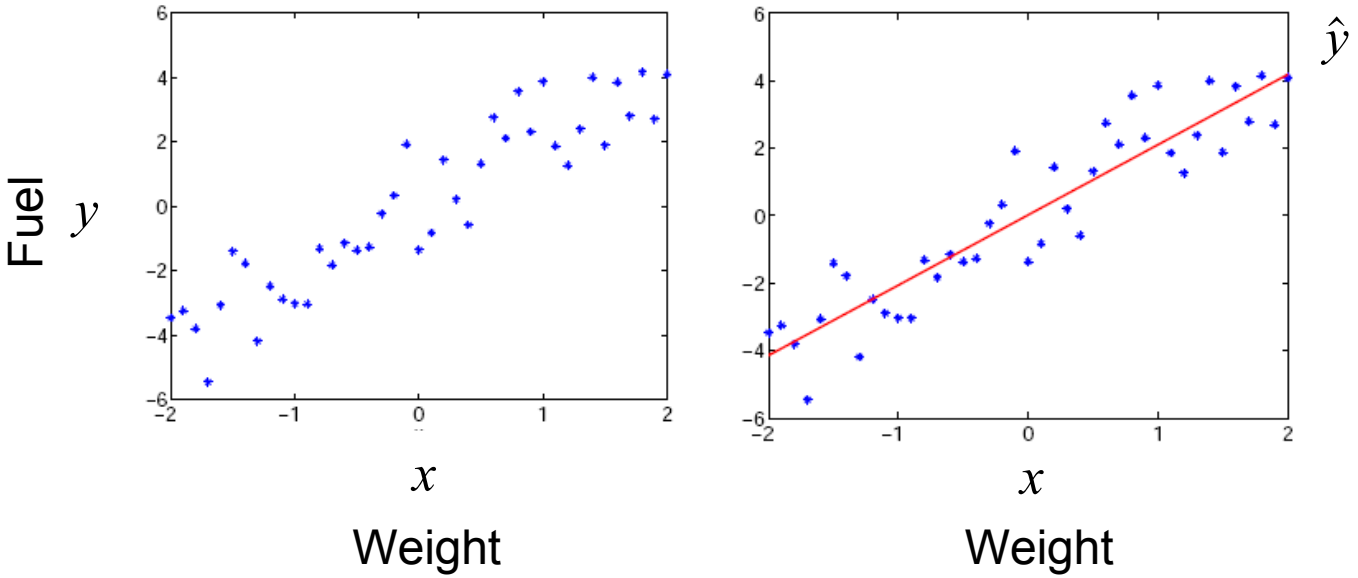
$$H : X \rightarrow Y \qquad \text{wobei } |Y| \text{ unendlich ist.}$$

Beispiel:
Vorhersage des Verbrauchs (in Liter) in Abhängigkeit von 8 Motorattributen.

y	x				
	cyls	disp	hp	weight	. . .
18.0	8	307.0	130.00	3504	. . .
26.0	4	97.00	46.00	1835	. . .
33.5	4	98.00	83.00	2075	. . .
. . .					

Einfachheitshalber verwenden wir nur ein Attribut.

Lineare Regression



Optimierung ist analytisch lösbar

Minimierung des empirischen Fehlers

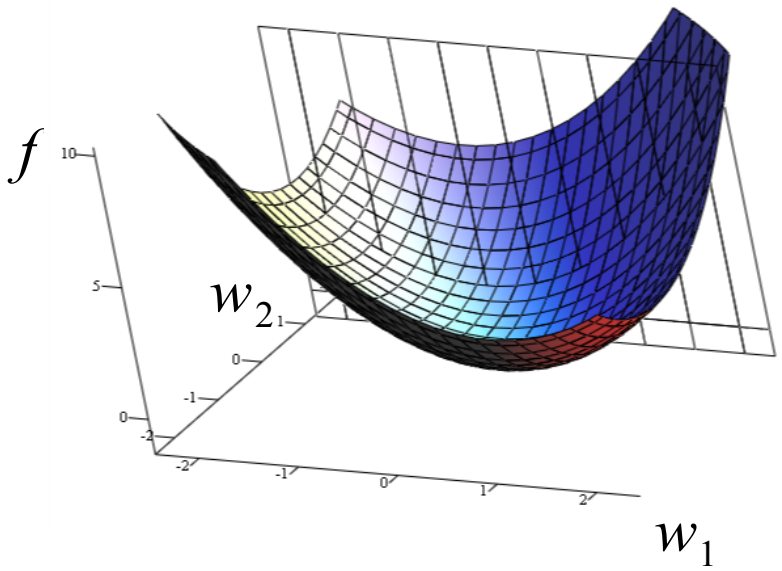
$$MSE(\boldsymbol{w}) = \frac{1}{l}\sum_{k=1}^{l}\left(y_k - H(x_k, \boldsymbol{w})\right)^2$$

Durch 0 setzen der Ableitung nach w_0 und w_i erhält man die „optimalen“ Parameterwerte

Einschub: Partielle Ableitungen

Partielle Ableitungen sind definiert als Ableitungen von Funktionen mehrerer Variablen $f(w_i, \ldots, w_d)$, wenn alle, außer der Variablen nach der abgeleitet wird, festgehalten werden

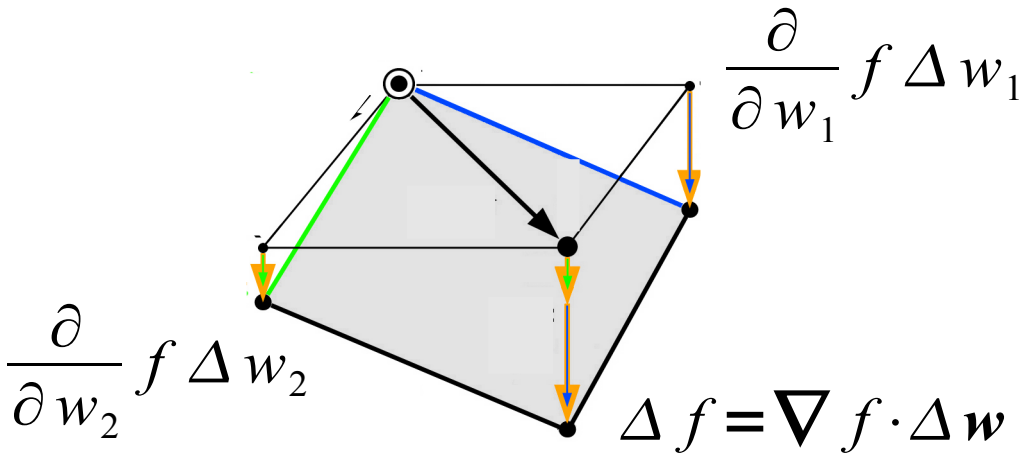
$$\frac{\partial}{\partial w_i} f \equiv \lim_{h \to \infty} \frac{f(w_1, .., w_i + h, .., w_d) - f(w_1, .., w_i, .., w_d)}{h}$$



Einschub: Partielle Ableitungen

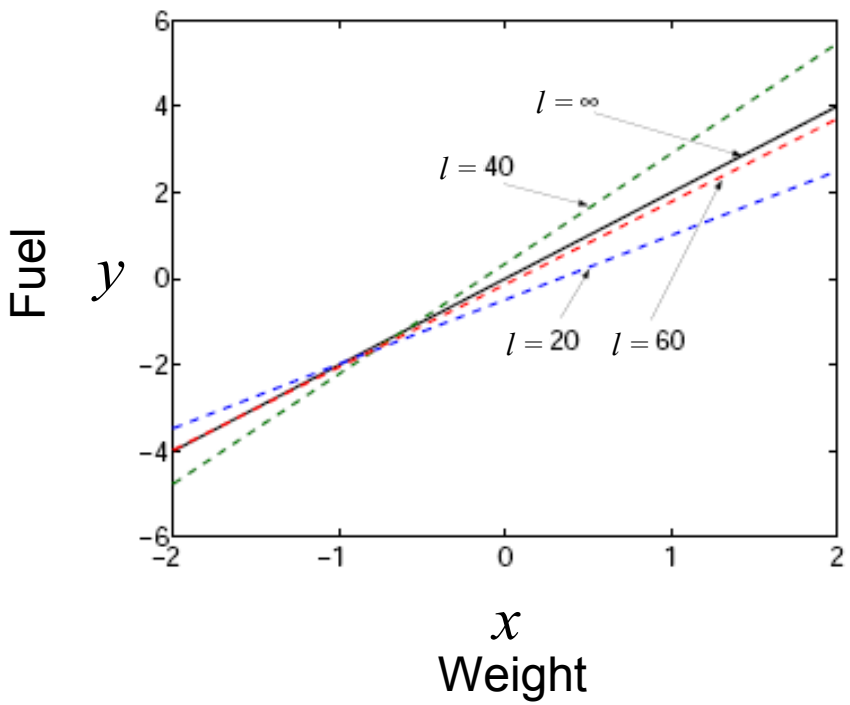
Es kann auch nach mehreren Variablen partiell abgeleitet werden.

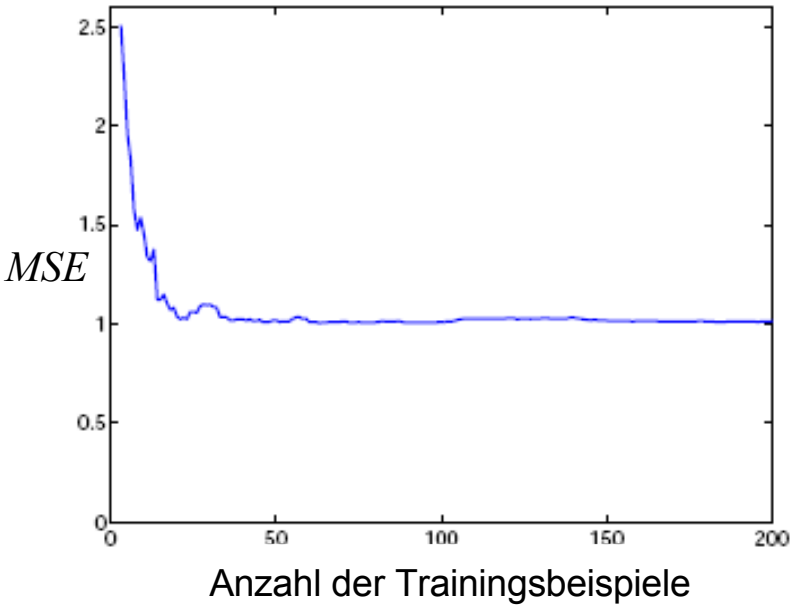
$$\nabla f \equiv \left(\frac{\partial f}{\partial w_1}, \ldots, \frac{\partial f}{\partial w_d} \right)$$



Vorhersagefehler

Je größer die Anzahl der Trainingsbeispiele, desto geringer der wahre Fehler.





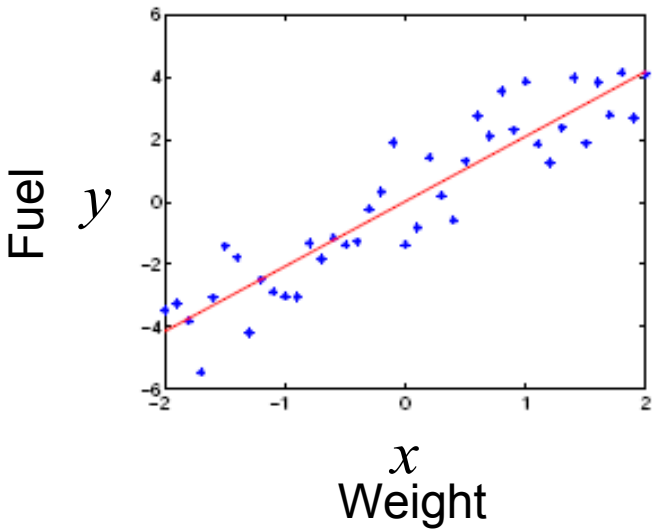
Wir versuchen den Fehler besser zu verstehen.

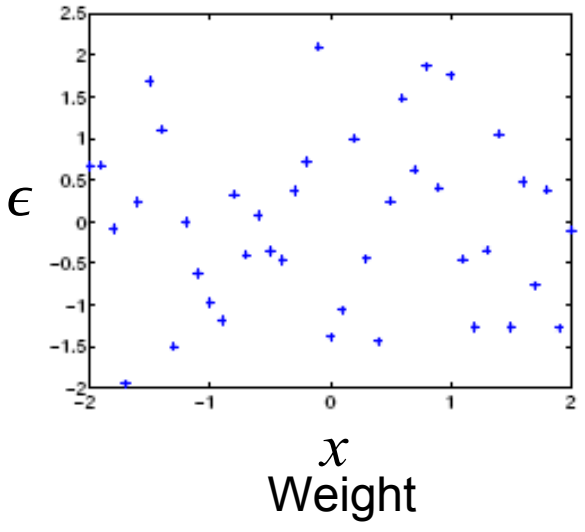
Fehleranalyse

Wenn wir den Vorhersagefehler mit $\epsilon_k = (y_k - w_0 - w_1 x_k)$ bezeichnen erhalten wir

$$\sum_{k=1}^{l} \epsilon_k (-x_k) = 0 \qquad \sum_{k=1}^{l} \epsilon_k = 0$$

Der Vorhersagefehler ist unkorreliert mit jeder linearen Funktion von x.





Fehleranalyse

Wenn wir den Vorhersagefehler mit $\epsilon_k = (y_k - w_0 - w_1 x_k)$ bezeichnen erhalten wir

$$\sum_{k=1}^{l} \epsilon_k (-x_k) = 0 \qquad \sum_{k=1}^{l} \epsilon_k = 0$$

Der Vorhersagefehler ist unkorreliert mit jeder linearen Funktion von x,

aber nicht mit anderen Funktionen, z.B. quadratischen Funktionen des inputs

$$\frac{2}{l} \sum_{k=1}^{l} \epsilon_k x_k^2 \neq 0 \qquad \text{(im Allgemeinen)}$$

Fehlerarten

- Der **strukurelle Fehler** misst den Fehler, der auf die limitierte Hypothesenklasse zurückzuführen ist (für unendlich viele Trainingsbeispiele)

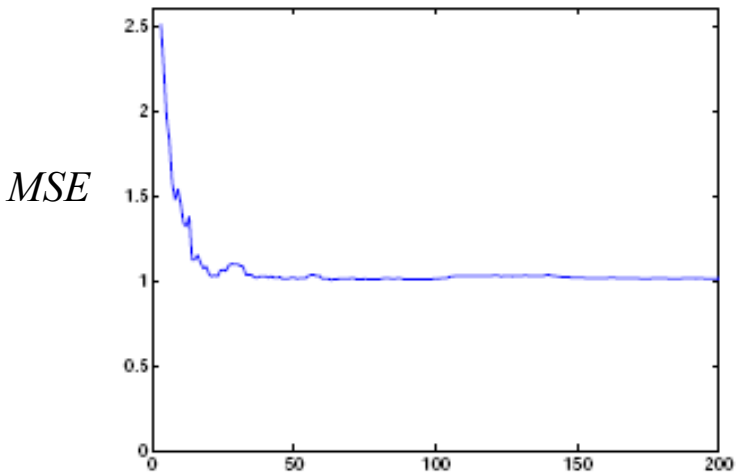
- Der **Approximationsfehler** gibt an, wie nahe man an der best möglichen Vorhersage liegt und ist zurückzuführen auf die geringe Anzahl von samples.

Fehlerdekomposition

Der erwartete Fehler der linearen Regressionsfunktion teilt sich in einen strukturellen Fehler und einen Approximationsfehler auf.

$$\left\langle (y - \hat{w}_0 - \hat{w}_1 x)^2 \right\rangle_{(\boldsymbol{x}, y) \sim P} = \left\langle (y - w_0^* - w_1^* x)^2 \right\rangle_{(\boldsymbol{x}, y) \sim P}$$

$$+ \left\langle (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right\rangle_{(\boldsymbol{x}, y) \sim P}$$



Fehlerdekomposition

$$\left\langle (y - \hat{w}_0 - \hat{w}_1 x)^2 \right\rangle_{(\boldsymbol{x}, y)\sim P}$$

$$= \left\langle \left| (y - w_0^* - w_1^* x) + (w_0^* + w_1^* x - \hat{w}_0 + \hat{w}_1 x) \right|^2 \right\rangle_{(\boldsymbol{x}, y)\sim P}$$

$$= \left\langle (y - w_0^* - w_1^* x)^2 \right\rangle_{(\boldsymbol{x}, y)\sim P}$$

$$+ \; 2 \left\langle (y - w_0^* - w_1^* x) \, (w_0^* + w_1^* x - \hat{w}_0 + \hat{w}_1 x) \right\rangle_{(\boldsymbol{x}, y)\sim P}$$

$$+ \left\langle (w_0^* + w_1^* x - \hat{w}_0 - \hat{w}_1 x)^2 \right\rangle_{(\boldsymbol{x}, y)\sim P}$$

Der zweite Term verschwindet da der Fehler mit jeder beliebigen affin-linearen Funktion des inputs unkorreliert ist.