Anwendungen der KI

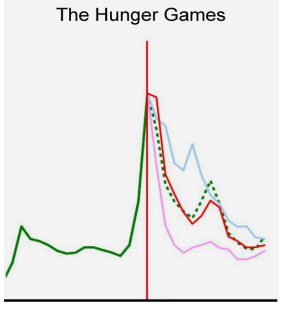
– Sommersemester 2018 –

Kapitel 05: Machine Learning

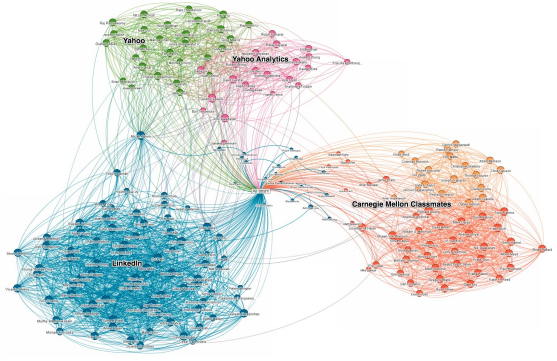
Prof. Dr. Adrian Ulges
B.Sc. Informatik (AI, ITS, MI, WI)
Fachbereich DCSM
Hochschule RheinMain

Machine Learning: Fragestellungen

Regression

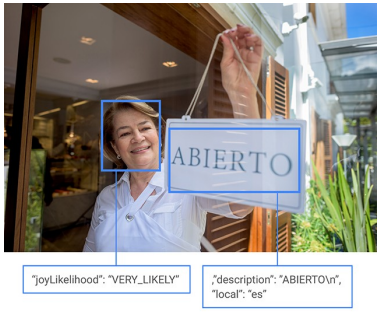


Clustering / Segmentierung



Recommendation

Klassifikation



Datenreduktion

Ausreißerdetektion

Begriffsklärung: Maschinelles Lernen

"Machine learning is the [...] study of **algorithms that can learn from data**. *Such algorithms operate by building a* **model** *from* **example inputs** *and using that to make* **predictions or decisions**."

(en.wikipedia.org)

"The field of study that gives computers the ability to learn **without being explicitly programmed.**"

(Arthur Samuel (1959))

"A computer program is said to **learn** *from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with Experience E."*

(Tom Mitchell (1998))

Machine Learning und Natural Language Processing

Wir verwenden Machine Learning, um aus **Beispieldaten** zu lernen, wie ein Wort/Satz/Dokument im Rahmen des gegebenen Kontexts zu **interpretieren** ist.

Beispiele

1. **Part-of-Speech (POS) Tagging**: Bestimmung der Wortart[1]

   "The **bear** *survives in summer on fish and fruit."* → *NN*
   "Your efforts will **bear** *fruit."* → *VB*

   (im Schnitt ca. 2 mögliche Wortarten pro Term, 92% Genauigkeit durch Wahl der häufigsten Wortart, 97% durch Machine Learning.)

2. **Sentiment-Analyse**

   "I can not believe it – What a cool video!" → ☺
   "This video is not cool – What a..." → ☹

3. **Antworttyp-Detektion**

   "Who founded Virgin Airlines" → *PERSON/INDIVIDUAL*
   "What state capital is the largest?" → *GEO/CITY*

[1] Englisch: 45 Wortarten im sog. *Penn Treebank Tagset*

Outline

Grundvokabular: Objekte, Merkmale

Stichprobe (engl. “Samples”)

- Das Ziel von Data Mining sind Aussagen / Prognosen über Objekte der Welt
  - **News-Classifier**: Objekte = News-Artikel
  - **Routenplaner**: Objekte = zu planende Fahrten
- Zum Lernen sammeln wir eine Stichprobe von Objekten und bezeichnen diese als **Samples**.

Merkmale (engl. “features”)

- Wir beschreiben jedes Sample anhand von **Merkmalen**
  - **News-Classifier**: Merkmale = Autor, Schlüsselterme, ...
  - **Routenplaner**: Merkmale = Tageszeit, Start, Ziel, ...
- Wir fassen alle Merkmale in einem **Merkmalsvektor x** zusammen

Grundvokabular: Objekte, Merkmale

Merkmale (cont'd)

- Merkmale können **qualitativ** *(Bsp. Farbe)* oder **quantitativ** *(Bsp. Alter)* sein.
- Wir können qualitative Merkmale mit **Indikator-Variablen** (engl. **One-Hot Encoding**) in quantitative überführen.

One-Hotc Encoding

	PS	Farbe
Auto von Prof. Ulges	70	weiß
Auto von Prof. Ulges' Frau	690	rot

PS	ist_grün	ist_silber	ist_rot
73	0	0	0
690	0	0	1

! Wir können Samples (meist) als **Punkte** in einem *(hochdimensionalen)* Raum interpretieren!

- Häufig: **Vorverarbeitung** der Daten
  - Schätzung fehlender Werte (engl. "**imputation**")
  - Verwerfen von Ausreißern
  - Vorauswahl der "wichtigen" Merkmale
  - **Standardisierung** der einzelnen Merkmale

Grundvokabular: (Un-)überwachtes Lernen

Überwachtes Lernen

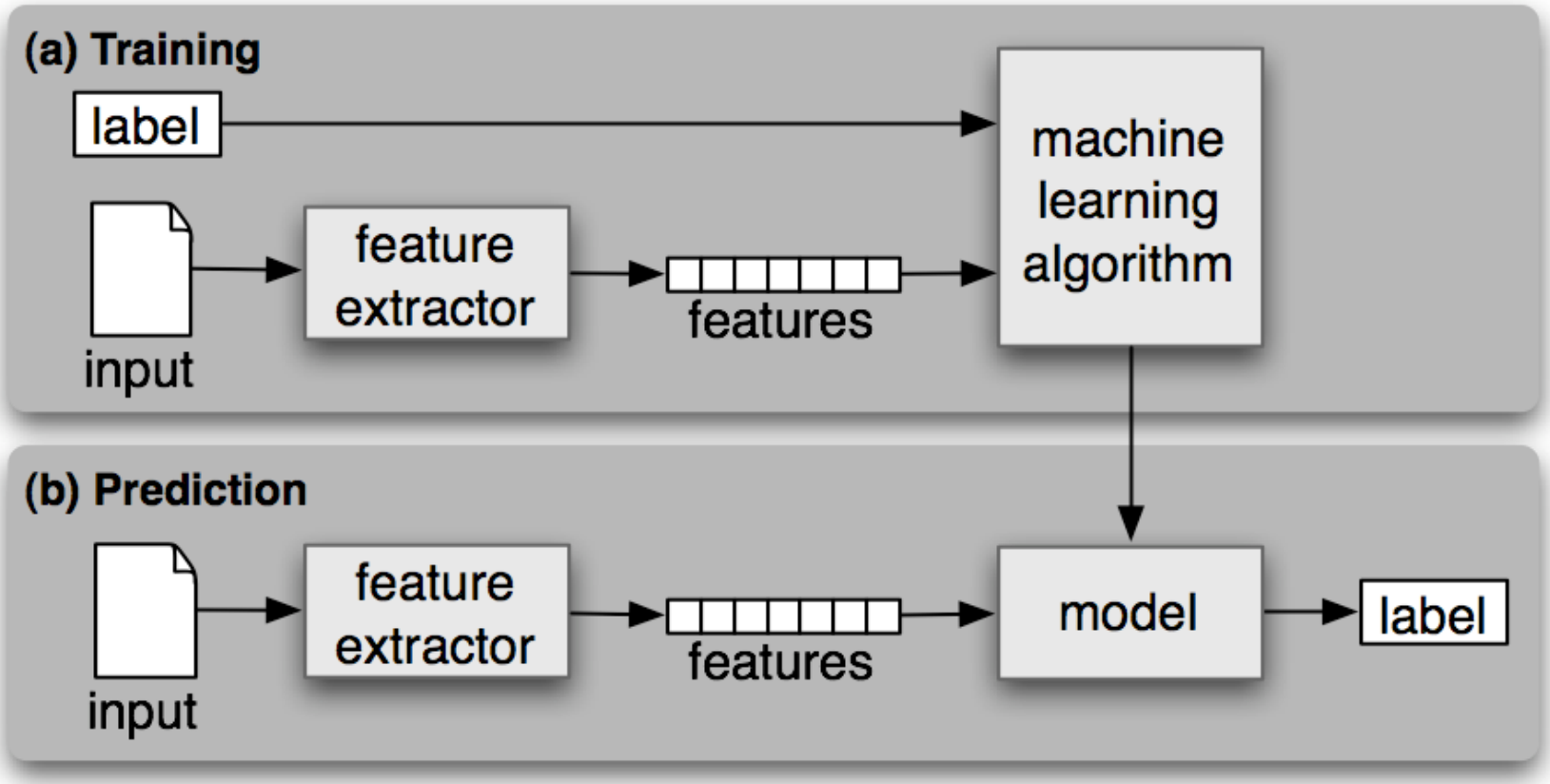
- Unser Ziel ist eine **Prognose** über ein Objekt. Wir bezeichnen den vorhergesagten Wert als **Label** oder **Target**.
- Lerne von Samples $\mathbf{x}_1, ..., \mathbf{x}_n$ und Labels $y_1, ..., y_n$
- Labels können **reellwertig** sein ($\rightarrow$ **Regressions-Problem**)
  - **Bsp: Routenplaner**: $y_i \in \mathbb{R}_0^+$ ($=$ *Zeit bis zum Ziel*)
- Labels können **nominal** sein, d.h. die Zugehörigkeit zu **Kategorien** ausdrücken ($\rightarrow$ **Klassifikations-Problem**).
  - **Bsp: Suchmaschine**: $y_i \in \{relevant, irrelevant\}$ (bzw. $\{0, 1\}$)

Unüberwachtes Lernen

- Unüberwachtes Lernen $=$ Lernen **ohne Labels**
- Lerne von Samples $\mathbf{x}_1, ..., \mathbf{x}_n$
- **Aufgabe**: Lerne die **Struktur** der Daten: Untergruppen, häufige Muster, Ausreißer/Anomalien

Überwachtes Lernen: (Batch-)Pipeline[2]

1. Wir **trainieren** das System in einer Offline-Phase und erhalten ein **Modell**
2. Wir wenden das Modell in der **Online-Phase** an



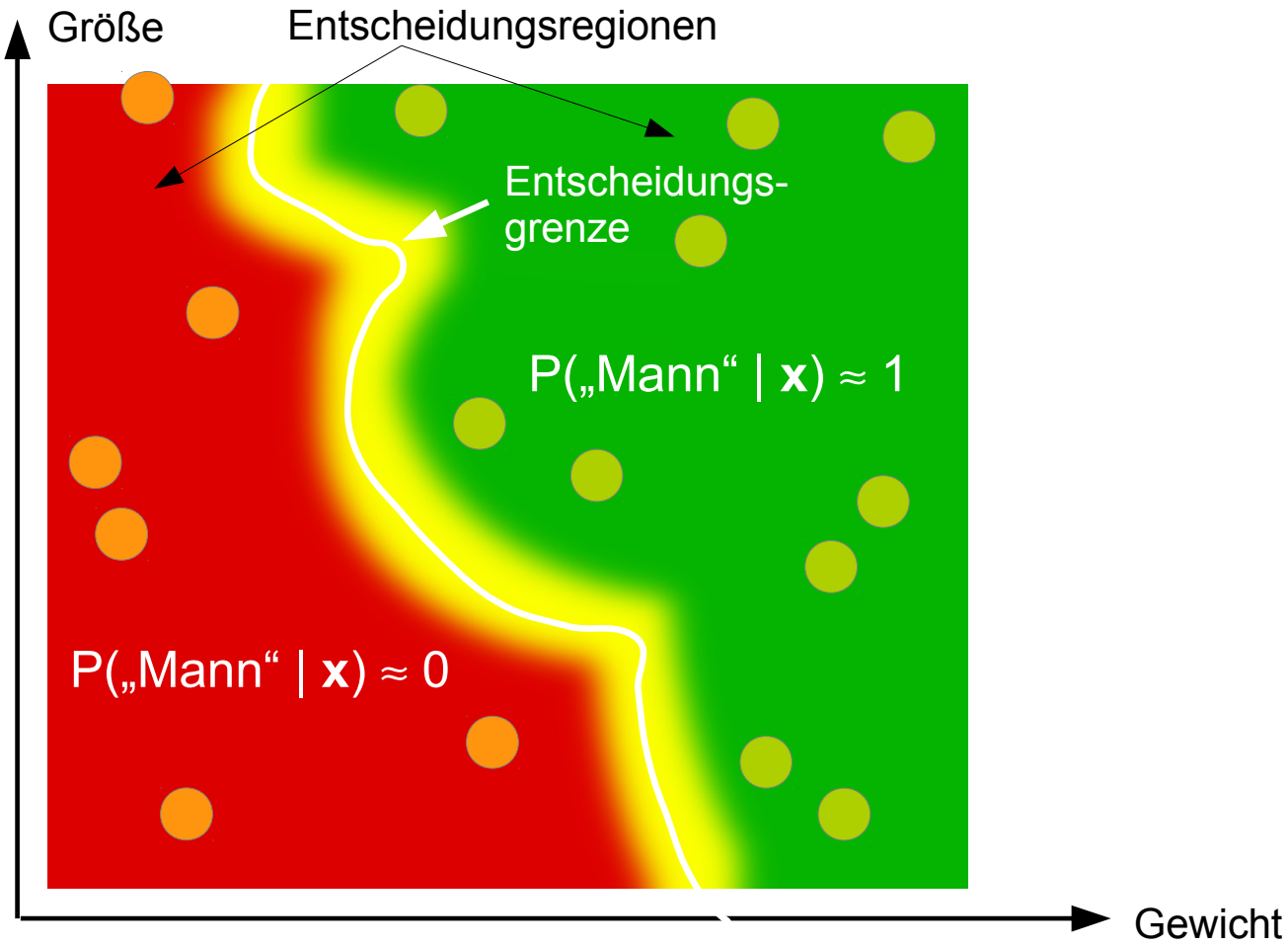
[2]Bildquelle: http://www.99designs.com

Machine Learning: Eingabedaten

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2. 3101282	7925		S
5	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy, Mr. Timothy J	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, Master. Gosta Leonard	male	2	3	1	349909	21075		S
10	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, Miss. Elizabeth	female	58	0	0	113783	26.55	C103	S
14	13	0	3	Saundercock, Mr. William Henry	male	20	0	0	A/5. 2151	8.05		S
15	14	0	3	Andersson, Mr. Anders Johan	male	39	1	5	347082	31275		S
16	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	350406	7.8542		S
17	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	248706	16		S
18	17	0	3	Rice, Master. Eugene	male	2	4	1	382652	29125		Q
19	18	1	2	Williams, Mr. Charles Eugene								
20	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Van								
21	20	1	3	Masselmani, Mrs. Fatima								
22	21	0	2	Fynney, Mr. Joseph J								
23	22	1	2	Beesley, Mr. Lawrence								
24	23	1	3	McGowan, Miss. Anna "Annie"								
25	24	1	1	Sloper, Mr. William Thompson								
26	25	0	3	Palsson, Miss. Torborg Danira								
27	26	1	3	Asplund, Mrs. Carl Oscar (Selma Augusta Emi								
28	27	0	3	Emir, Mr. Farred Chehab								
29	28	0	1	Fortune, Mr. Charles Alexander								
30	29	1	3	O'Dwyer, Miss. Ellen "Nellie"								
31	30	0	3	Todoroff, Mr. Lalio								
32	31	0	1	Uruchurtu, Don. Manuel E								
33	32	1	1	Spencer, Mrs. William Augustus (Marie Eugen								
34	33	1	3	Glynn, Miss. Mary Agatha								
35	34	0	2	Wheadon, Mr. Edward H								
36	35	0	1	Meyer, Mr. Edgar Joseph								
37	36	0	1	Holverson, Mr. Alexander Oskar								
38	37	1	3	Mamee, Mr. Hanna								
39	38	0	3	Cann, Mr. Ernest Charles								
40	39	0	3	Vander Planke, Miss. Augusta Maria								
41	40	1	3	Nicola-Yarred, Miss. Jamila								
42	41	0	3	Ahlin, Mrs. Johan (Johanna Persdotter Larsso								
43	42	0	2	Turpin, Mrs. William John Robert (Dorothy Ann								
44	43	0	3	Kraeff, Mr. Theodor								

Machine Learning: Eingabedaten (geometrisch)

- **Szenario**: Klassifikation (ordne Sample **x** einer Klasse c zu)
- **Ziel**: Berechne $P(c|\mathbf{x})$ für jede Klasse c.
- Hieraus ergeben sich **Entscheidungsregionen** und -**Grenzen**.



Outline

Logistische Regression: Ansatz

- **Logistische Regression** ist ein einfaches, weit verbreitetes Modell zur Klassifikation.
- Gegeben: Trainingssamples $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ mit zugehörigen Labels $y_1, ..., y_n$ (Klassen-Zugehörigkeit).
- Ziel: Lerne eine **Klassifikator**-Funktion $\mathbb{R}^d \to \{0, 1\}$, die den Samples eine **Klasse C** zuordnet *(erstmal nur 2 mögliche Klassen: Erfolg/Misserfolg, gesund/krank, Spam/Ham!)*.

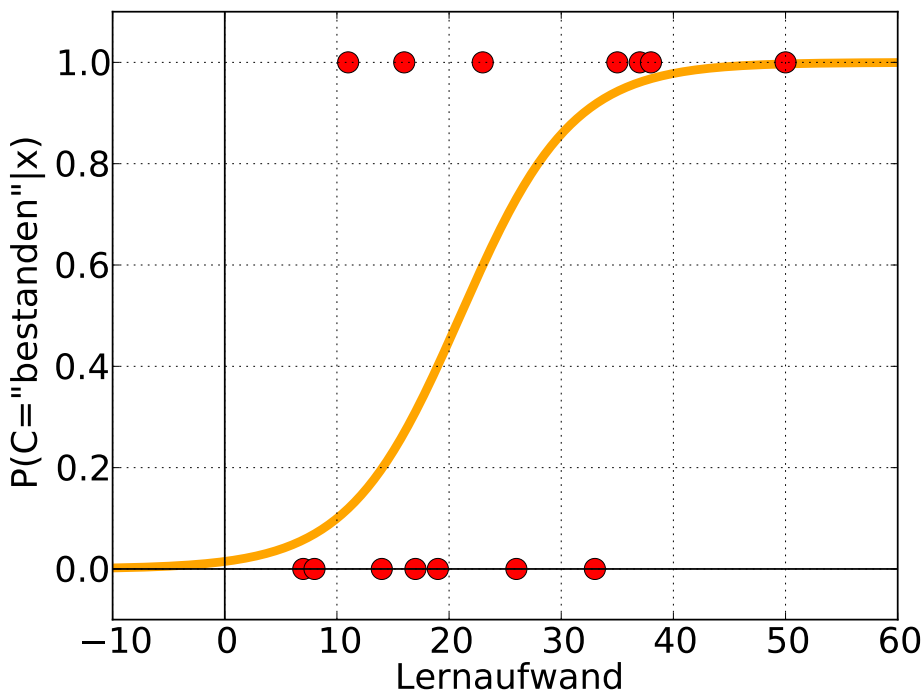
Der Ansatz...

- ... ist **probabilistisch**: Unser Modell schätzt $P(C = 1|\mathbf{x})$.
- Der Klassifikator entscheidet sich für die Klasse mit **maximaler Wahrscheinlichkeit**.

Logistische Regression: Ansatz

Beispiel: Mathe-Klausur

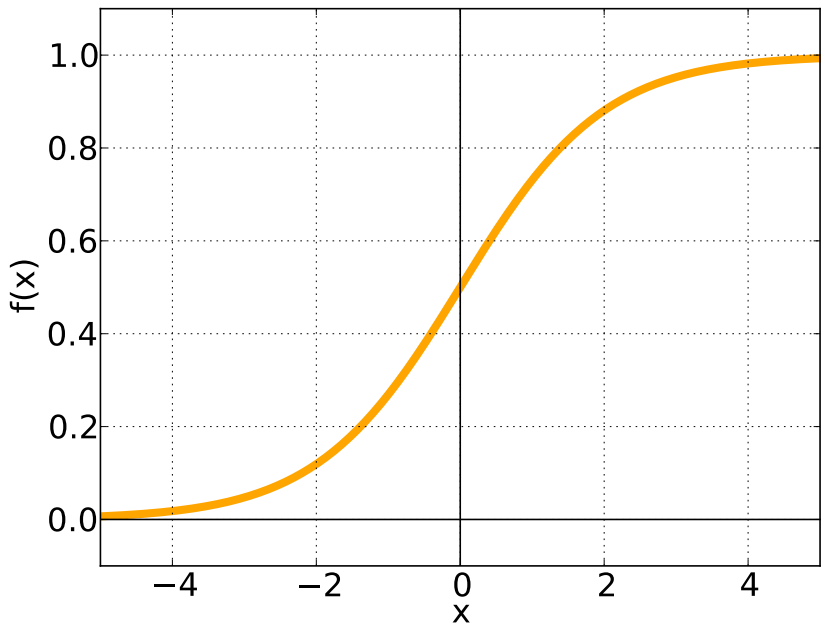
- x = Lernaufwand, C = bestanden / nicht bestanden
- **Gegeben**: Eine Trainingsmenge $x_1, x_2, ..., x_n \in \mathbb{R}$ mit Labels $y_1, y_2, ..., y_n \in \{0, 1\}$.
- **Ziel**: Ermittle $P(C = 1|x)$ für ein Test-Sample x



Logistische Regression: Grundmodell

- Wir verwenden für die Regressionsfunktion als **Grundmodell** die sogenannte **Sigmoid-Funktion**

$$P(C = 1|x) := f(x) = \frac{1}{1 + e^{-x}}$$



- Es gilt: $lim_{x \to -\infty} f(x) = 0$ und $lim_{x \to \infty} f(x) = 1$
- Es gilt: $P(C = 1|x = 0) = f(0) = 0.5$.
  Wir entscheiden uns also für Klasse 1, falls $x \geq 0$.

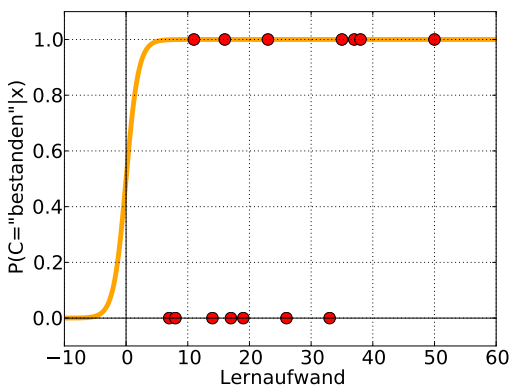
Logistische Regression: Grundmodell

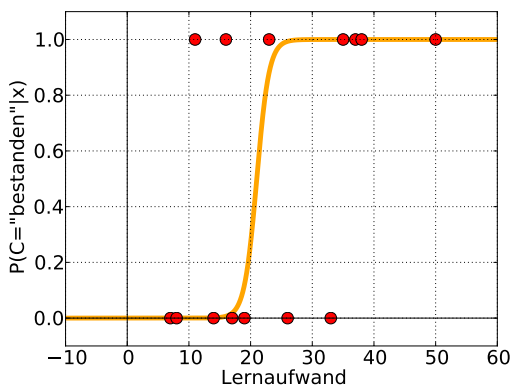
Erweiterung

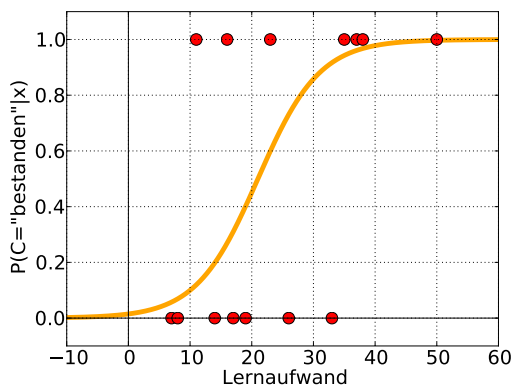
- Wir erlauben eine **Verschiebung** und **Streckung/Stauchung/Spiegelung** der Funktion!
- Wir erhalten als Modell:

$$f(x; w_0, w) = \frac{1}{1 + e^{-(w_0 + w \cdot x)}}$$

- Die Parameter w_0, w werden beim **Training** ermittelt *(gleich)*.







Logistische Regression: Anmerkungen

Warum dieses Modell?

- **Einfachheit**, Intuition.
- Das Modell ist korrekt für **normalverteilte** Klassen gleicher Varianz.
- **Tradition**.
- **Wenige Parameter** zu fitten → Gute Ergebnisse bereits bei wenigen Trainingssamples.

Würde **lineare** Regression funktionieren? → Nein, denn ...

- ... wir können durch die Daten keine sinnvolle Gerade fitten
- ... die prognostizierten Wahrscheinlichkeiten lägen nicht zwischen 0 und 1

Logistische Regression im Mehrdimensionalen

- Wie modellieren wir **multivariate** Samples $\mathbf{x} \in \mathbb{R}^d$?
- Wir erweitern die Sigmoid-Funktion:

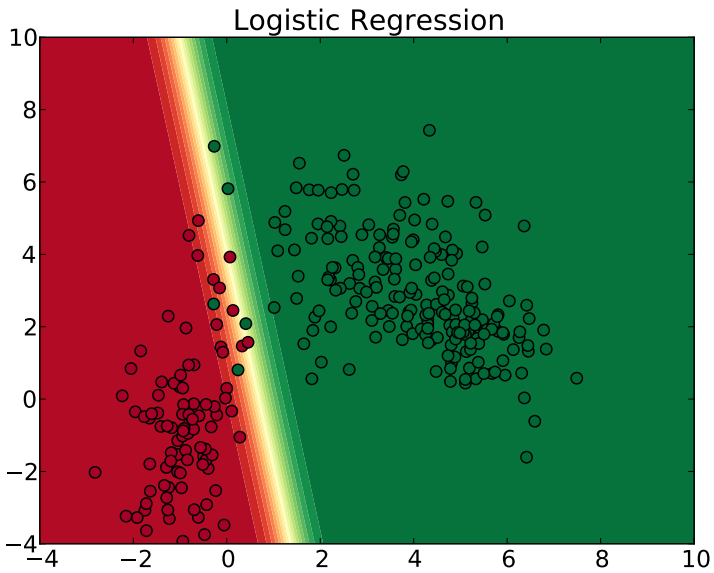
$$f(\mathbf{x}; w_0, w_1, w_2 \ldots, w_d) = \frac{1}{1 + e^{-(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_d \cdot x_d)}}$$

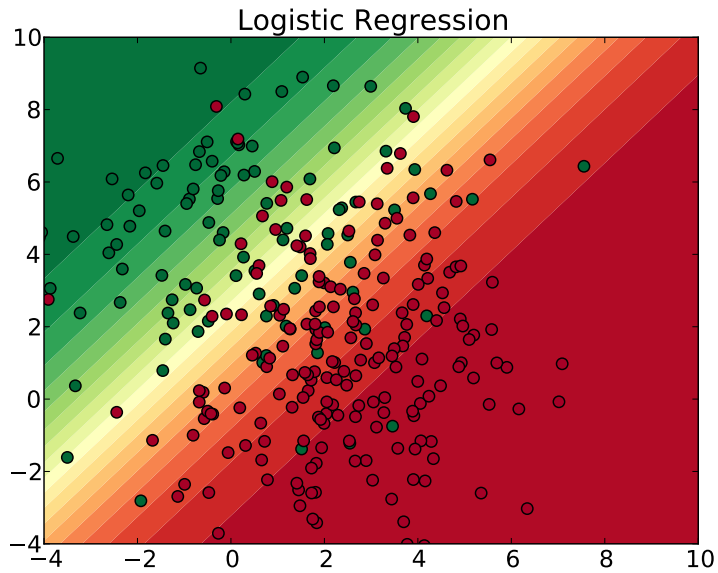
oder kurz (mit Vektor $\boldsymbol{w} := (w_1, \ldots, w_d)$):

$$f(\mathbf{x}; w_0, \boldsymbol{w}) = \frac{1}{1 + e^{-(w_0 + \mathbf{x} \cdot \boldsymbol{w})}}$$

- Die Entscheidungsgrenze dieses Modells liegt bei $\boldsymbol{x} \cdot \boldsymbol{w} + w_0 = 0$. Dies entspricht einer **Hyperebene** (in Normalenform)!

Logistische Regression: Illustration





- Die Entscheidungsgrenze ist linear. Wir sprechen bei logistischer Regression deshalb auch von einem **linearen Klassifikator** *(es gibt noch einige weitere!)*.
- Der Parameter $\boldsymbol{w}$ bestimmt die Orientierung der Entscheidungsgrenze, w_0 verschiebt die Grenze.
- $\boldsymbol{w}$ bestimmt darüber hinaus die **Glätte** der Entscheidungsfunktion f.

Logistische Regression: Training

Schlüsselfrage: Training

- Gegeben ist eine Trainingsmenge von Samples $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ mit Labels $y_1, ..., y_n \in \{0, 1\}$
- **Ziel**: Bestimme w_0 und $\mathbf{w}$, also die Position der Entscheidungsgrenze und Glätte der Entscheidungsfunktion

Ansatz: Maximum-likelihood-Schätzung

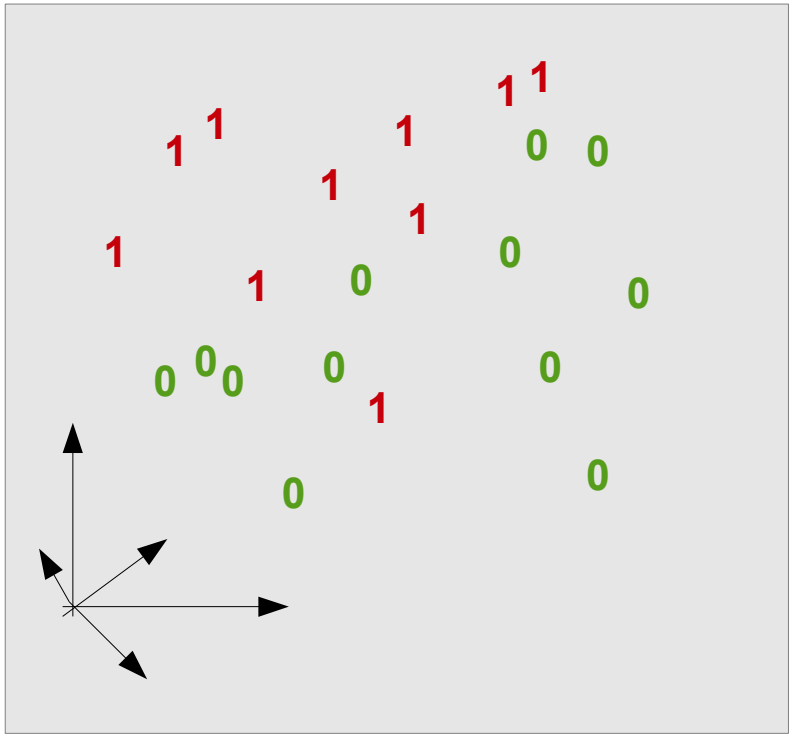
- **Grundidee**: Wähle die Parameter so, dass die beobachten Daten "maximal wahrscheinlich" werden
- Für **positive** Samples ($y_i = 1$) sollte $f(\mathbf{x}_i)$ möglichst **groß** sein:

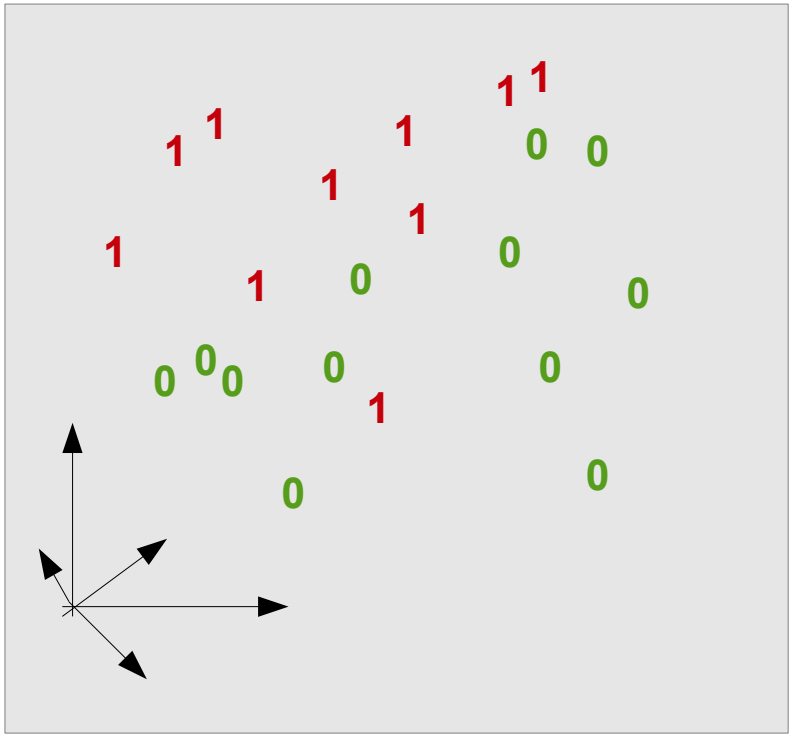
$$\Big(P(C = 1|\mathbf{x}_i)\Big) \approx f(\mathbf{x}_i) \approx 1$$

- Für **negative** Samples ($y_i = 0$) sollte $f(\mathbf{x}_i)$ möglichst **klein** sein:

$$\Big(P(C = 1|\mathbf{x}_i)\Big) \approx f(\mathbf{x}_i) \approx 0$$

Logistische Regression: Beispiel





Logistische Regression: Ansatz

Maximum-Likelihood-Schätzung

Wir formulieren eine sogenannte Likelihood-Funktion und stellen ein Maximierungsproblem auf:

$$w_0^*, \mathbf{w}^* = \arg\max_{w_0,\mathbf{w}} \underbrace{\prod_{i:y_i=1} f(\mathbf{x}_i) \cdot \prod_{i:y_i=0} (1 - f(\mathbf{x}_i))}_{\text{"Likelihood-Funktion" } L(w_0,\mathbf{w})}$$

Wir formen das Maximierungsproblem um:

$$\begin{aligned} w_0^*, \mathbf{w}^* &= \arg\max_{w_0,\mathbf{w}} \prod_{i:y_i=1} f(\mathbf{x}_i) \cdot \prod_{i:y_i=0} (1 - f(\mathbf{x}_i)) \\ &= \arg\max_{w_0,\mathbf{w}} \prod_{i} f(\mathbf{x}_i)^{y_i} \cdot (1 - f(\mathbf{x}_i))^{1-y_i} && // \log \\ &= \arg\max_{w_0,\mathbf{w}} \sum_{i} y_i \cdot log(f(\mathbf{x}_i)) + (1 - y_i) \cdot log(1 - f(\mathbf{x}_i)) \end{aligned}$$

Logistische Regression: Ansatz

$$
\begin{aligned}
w_0^*, \mathbf{w}^* &= \arg\max_{w_0,\mathbf{w}} \underbrace{\prod_{i:y_i=1} f(\mathbf{x}_i) \cdot \prod_{i:y_i=0} (1 - f(\mathbf{x}_i))}_{\text{"Likelihood-Funktion" } L(w_0,\mathbf{w})} \\
&= \arg\max_{w_0,\mathbf{w}} \prod_i f(\mathbf{x}_i)^{y_i} \cdot (1 - f(\mathbf{x}_i))^{1-y_i} && // \ \log \\
&= \arg\max_{w_0,\mathbf{w}} \sum_i y_i \cdot log(f(\mathbf{x}_i)) + (1 - y_i) \cdot log(1 - f(\mathbf{x}_i)) \\
&= \arg\max_{w_0,\mathbf{w}} \sum_i log(1 - f(\mathbf{x}_i)) + y_i \cdot log\Big(\frac{f(\mathbf{x}_i)}{1 - f(\mathbf{x}_i)}\Big) \\
&= \arg\max_{w_0,\mathbf{w}} \sum_i log\Big(\frac{\cancel{1} + exp(-(w_0 + \mathbf{x}_i\mathbf{w})) - \cancel{1}}{1 + exp(-(w_0 + \mathbf{x}_i\mathbf{w}))}\Big) + y_i \cdot log\left(\frac{\frac{1}{\cancel{(1+exp(-(w_0+\mathbf{x}_i\mathbf{w})))}}}{\frac{exp(-(w_0+\mathbf{x}_i\mathbf{w}))}{\cancel{(1+exp(-(w_0+\mathbf{x}_i\mathbf{w})))}}}\right) \\
&= \arg\max_{w_0,\mathbf{w}} \sum_i -log\Big(\frac{1 + exp(-(w_0 + \mathbf{x}_i\mathbf{w}))}{exp(-(w_0 + \mathbf{x}_i\mathbf{w}))}\Big) - y_i \cdot \cancel{log\Big(exp(-(w_0 + \mathbf{x}_i\mathbf{w}))\Big)} \\
&= \arg\max_{w_0,\mathbf{w}} \sum_i -log\Big(e^{w_0+\mathbf{x}_i\mathbf{w}} + 1\Big) + \sum_i y_i \cdot (w_0 + \mathbf{x}_i\mathbf{w})
\end{aligned}
$$

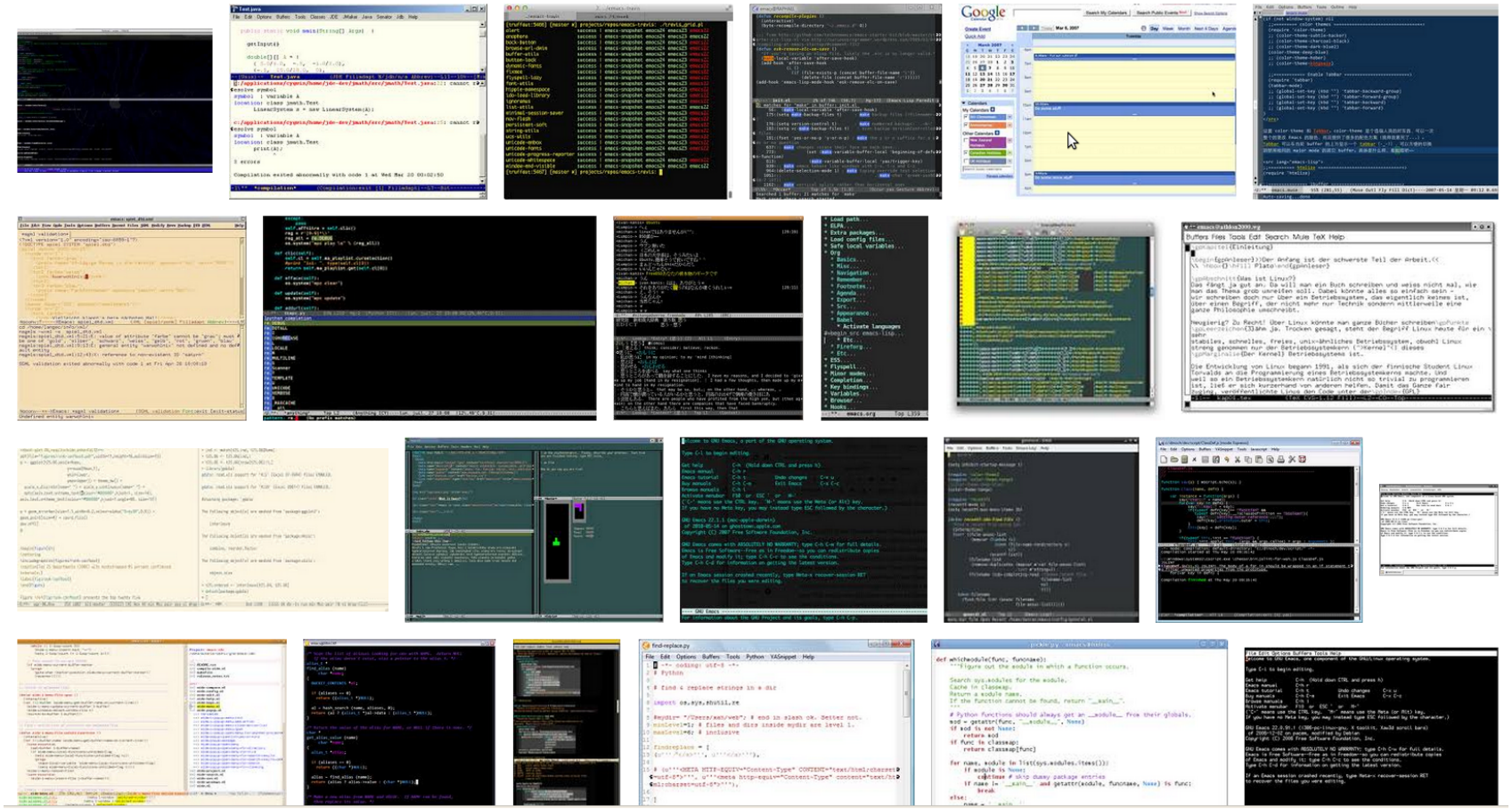
Logistische Regression: Ansatz

$$\arg\max_{w_0,\mathbf{w}} \underbrace{\sum_i -log\left(e^{w_0+\mathbf{x}_i\mathbf{w}}+1\right)+\sum_i y_i\cdot(w_0+\mathbf{x}_i\mathbf{w})}_{\text{"Log-Likelihood-Funktion" } l(w_0,\mathbf{w})}$$

Anmerkungen

- Diese Log-Likelihood-Funktion ist nicht analytisch minimierbar. Es gibt aber **numerische Lösungen**: Finde z.B. Nullstellen der Ableitung mit Hilfe des **Newton-Verfahrens**.
- Die Gewichte in $\mathbf{w}$ zeigen die **Bedeutung** der einzelnen Merkmale für das Klassifikationsproblem an.

Logistische Regression: Code-Beispiel



- Bag-of-Words Features
- Logistische Regression
- Inspektion der Merkmals-Gewichte

Logistische Regression: Regularisierung

- **Beobachtung**: Das gelernte Modell tendiert zum **Overfitting**, wenn ...
  - ... einzelne Merkmale zu hohe Gewichte erhalten
  - ... viele unwichtige Merkmale ein Gewicht $\neq 0$ erhalten
- Deshalb **regularisieren** wir das Problem, so dass die Einträge in $\mathbf{w}$ eher klein (bzw. null) sind.
- Wir definieren eine **Norm** des Gewichtsvektors $\mathbf{w}$

$$\|\mathbf{w}\|_1 := |w_1| + |w_2| + \ldots + |w_d| \qquad \text{L1-Norm}$$

$$\|\mathbf{w}\|_2 := \sqrt{w_1^2 + w_2^2 + \ldots + w_d^2} \qquad \text{L2-Norm}$$

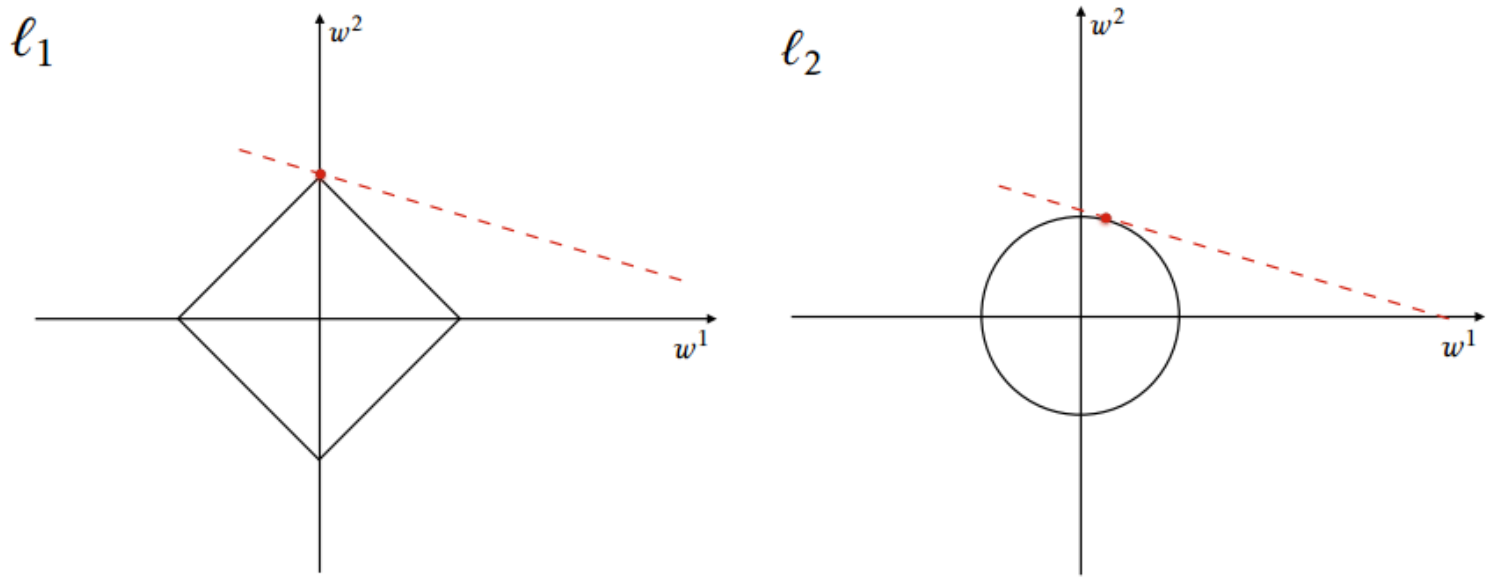
- Wir passen das Optimierungsproblem an, so dass hohe Gewichte in $\mathbf{w}$ "bestraft" werden:

$$\arg\max_{w_0,\mathbf{w}} \; l(w_0, \mathbf{w}) - C \cdot \|\mathbf{w}\|_1 \qquad // \text{ L1-Regularisierung}$$

$$\arg\max_{w_0,\mathbf{w}} \; l(w_0, \mathbf{w}) - C \cdot \|\mathbf{w}\|_2 \qquad // \text{ L2-Regularisierung}$$

Logistische Regression: Regularisierung

Welchen Unterschied macht der Typ der Regularisierung?



- Im Bild: $\mathbf{w} = (0, 1)$ (= L1-Lösung)
  vs. $\mathbf{w} = (0.15, 0.99)$ (=L2-Lösung)
- L1-Regularisierung setzt die Gewichte uninformativer Features auf 0 (engl. "sparsity"). Das heißt, der Klassifikator führt eine interne **Merkmalsselektion** durch!
- L2-Regularisierung reduziert Ausreißer *(= extreme Gewichte)*.

Logistische Regression: Fazit

- Logistische Regression ist ein sehr **einfaches** Modell
- Nur lineare Entscheidungsgrenzen sind modellierbar!
- **Grundprinzip**: Weise jedem Merkmal ein **Gewicht** zu.
- Klappt schon mit relativ **wenigen Trainingssamples** gut.
  Faustregel: ca. 10 Trainingssamples pro Klasse pro Merkmal.

Outline

Klassifikation: Illustration im Merkmalsraum

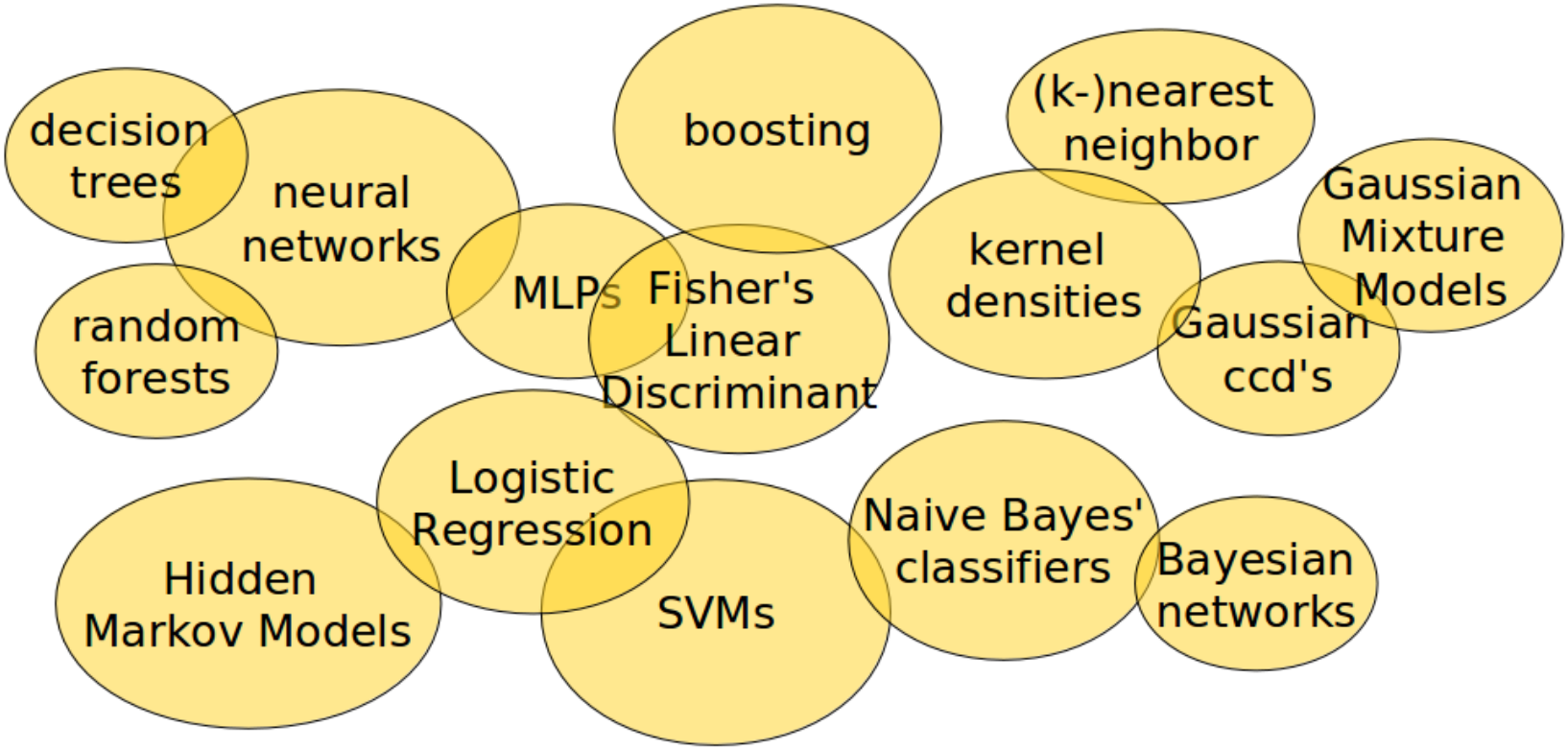
Generalisierung / Overfitting

"The real value of a scientific explanation lies not in its ability to explain (what one has already seen), but in predicting events that have yet to (be seen)."

(Blumer et al. 1987)

- Die Fähigkeit der eines Klassifikators, von den Trainingsdaten auf (neue) Testdaten zu schließen, bezeichnen wir als **Generalisierung**.
- Funktioniert ein Klassifikator sehr gut auf den Trainingsdaten, aber schlecht auf neuen Daten, sprechen wir von **Overfitting**.
- Overfitting tritt auf, wenn...
  - ... die Trainingsmenge sehr klein ist
  - ... das Modell sehr viele Parameter hat
  - ... das Modell nicht gut auf die Daten passt

Maschinelles Lernen: Theorie

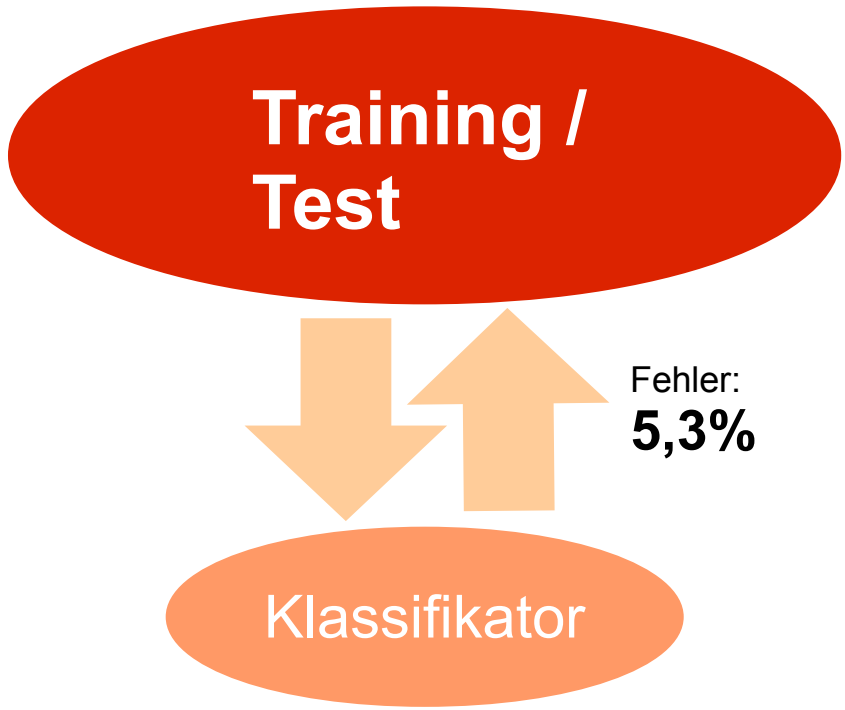


Was wir über maschinelles Lernen wissen

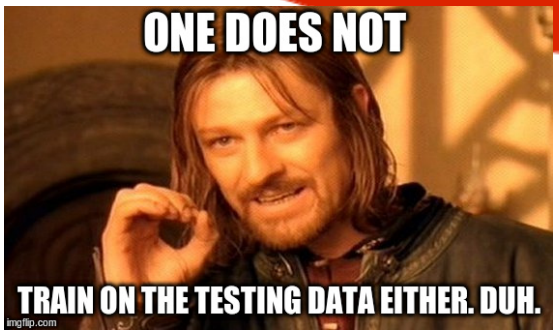
- Es gibt **zahlreiche** Methoden / Modelle.
- Es gibt kein **universell bestes** Modell *(no-free-lunch theorem)*.

Evaluation von ML-Systemen

- **Machine Learning** in der Praxis = iterative Auswahl von...
  - Daten
  - Merkmalen
  - Modellen
  - Parametern
- **Schlüsseltreiber**: **Benchmarking**



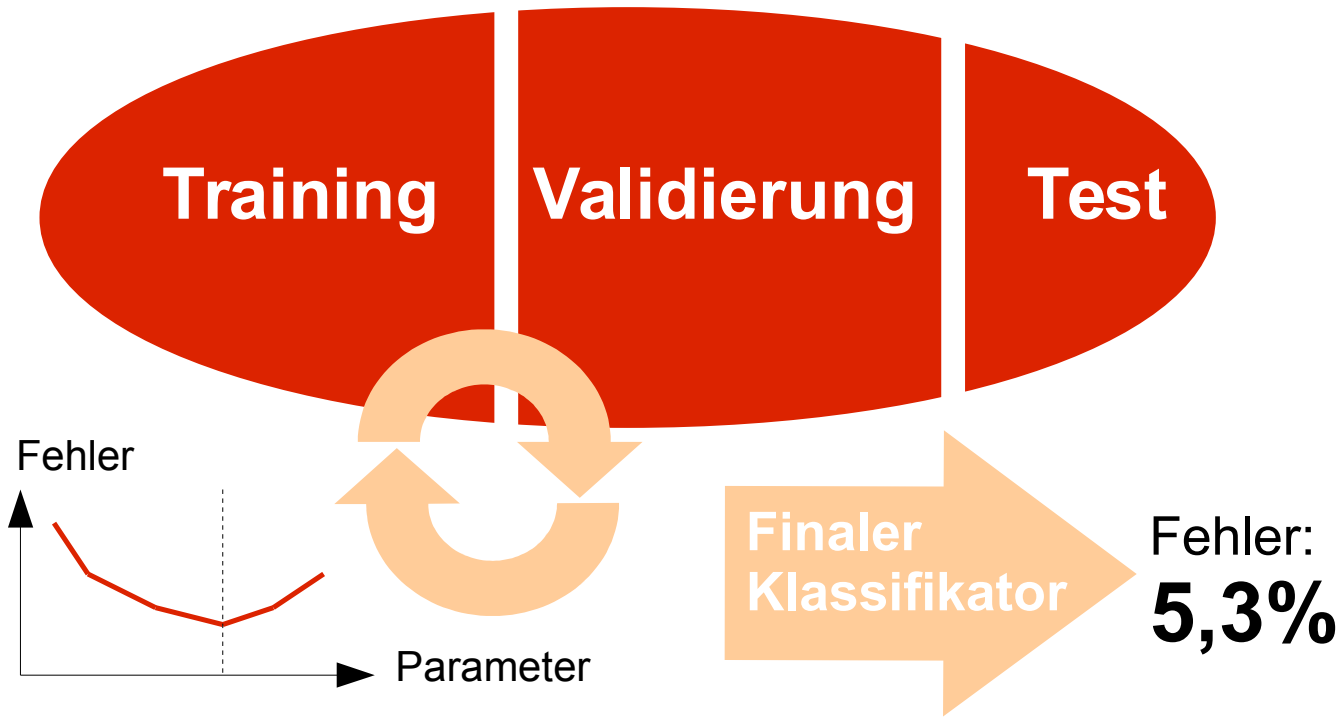
Data Mining: Benchmarking



Fehler:
5,3%

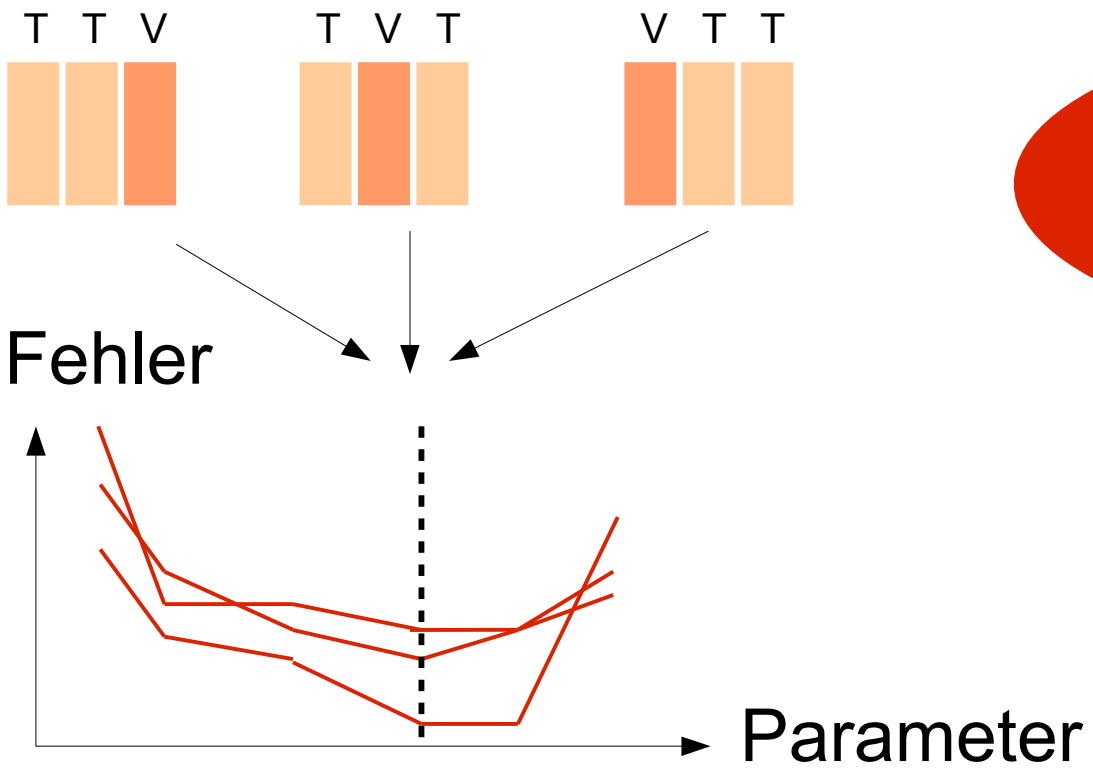
- Wir unterteilen die Daten in **Trainings- und Testdaten**.
- Wir **benchmarken** (nur) auf den Testdaten.
- **Todsünde**: “Testing on the Training Data”!
- **Todsünde (auch)**: “Training on the Test Data”!

Data Mining: Benchmarking



- Einige Parameter des Klassifikators werden **gelernt**, andere **(freie)** Parameter werden **manuell** gesetzt.
- Typischer Ansatz: **Manuelle Suche** / **Grid Search**.
- **Beispiel** Logistische Regression: Stärke der Regularisierung C.
- **Ansatz**: Training → Validieren → Testen.

Data Mining: Benchmarking



training + validation

Finaler Klassifikator

- Sind die **Trainingsdaten klein**, teilt man sie in Teile (engl. "**Folds**") und trainiert/validiert **mehrfach**.
- Die Ergebnisse werden über die Folds gemittelt.

Outline

Logistische Regression im NLP

Wir nehmen zwei Änderungen an logistischer Regression vor:

1. Merkmalsfunktionen
   - Wir verwenden statt Merkmalsvektoren **x** Merkmalsfunktionen $f_1, ..., f_n$. Diese nehmen unsere Merkmalsextraktion vor.
   - Eine **Merkmalsfunktion** f_i berechnet – gegeben Eingabewort/Eingabetext x und Klasse c – einen Merkmalswert $f_i(x, c)$.
   - Häufig sind die Merkmalswerte binär *(sogenannte* **Indikatorfunktionen***)*.

Beispiele

- Sentiment-Klassifikation

$$f_i(x,c) = \begin{cases} 1 & \text{falls } (\text{"great"} \in x \\ & \quad \wedge\ c = +) \\ 0 & \text{sonst} \end{cases}$$

- Satzzeichen-Erkenner: Liegt Satzende (EOS) vor?

$$f_i(x,c) = \begin{cases} 1 & \text{falls } (case(w_{i+1}) = upper \\ & \quad \wedge\ c = EOS) \\ 0 & \text{sonst} \end{cases}$$

Logistische Regression im NLP

2. Behandlung von mehr als zwei Klassen

- Wir ändern die Formel zur logistischen Regression leicht ab und erhalten die **multinomielle** logistische Regression

$$P(c|x) = \frac{exp\Big(\sum_i w_i \cdot f_i(x,c)\Big)}{\sum_{c'} exp\Big(\sum_i w_i \cdot f_i(x,c')\Big)}$$

- **Idee**: Nicht mehr ein exp-Term, der zwischen zwei Klassen entscheidet, sondern ein exp-Term für **jede Klasse**!

Logistische Regression im NLP: Beispiel [1] (Kapitel 7)

Merkmalsfunktionen

$$f_1(x,c) = \mathbf{1}_{\text{'great'} \in x \wedge c=+}$$

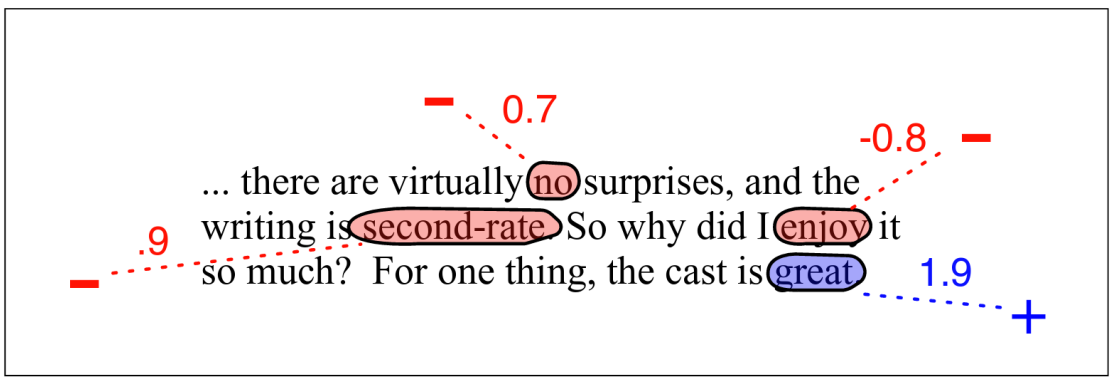
$$f_2(x,c) = \mathbf{1}_{\text{'no'} \in x \wedge c=-}$$

$$f_3(x,c) = \mathbf{1}_{\text{'second-rate'} \in x \wedge c=-}$$

$$f_4(x,c) = \mathbf{1}_{\text{'enjoy'} \in x \wedge c=-}$$

$$f_5(x,c) = \mathbf{1}_{\text{'chuck norris'} \in x \wedge c=+}$$

Gelernte Gewichte



Klassifikationsergebnis

$$P(c=+|x) = \frac{e^{1.9+0}}{e^{1.9+0} + e^{0.9+0.7-0.8}} = 82\%$$

$$P(c=-|x) = \frac{e^{0.9+0.7-0.8}}{e^{1.9+0} + e^{0.9+0.7-0.8}} = 18\%$$

Auswahl von Merkmalsfunktionen

Gute **Merkmalsfunktionen** hängen vom Klassifikationsproblem ab

- **Satzzeichen-Erkennung**: Groß/Kleinschreibung, Vorhandensein einer bekannten Abkürzungen *(U.S.A., St., ...)*
- **Spam-Klassifikation**: Identität des Absenders
- **Sentiment-Klassifikation**: Adektive, Bigramme *("very funny" vs. "not funny")*
- ...

Häufig treffen wir eine **Vorauswahl**

- Warum? Effizienzgründe, Overfitting
- Berechne für jedes Merkmal ein **Proxy-Maß**, das angibt ob eine Korrelation zur Klasse besteht (z.B. *information gain*)
- Ranke die Merkmale nach diesem Maß und verwende nur die **"besten" K Merkmalsfunktionen** (K: Tausende, ermittelt via Cross-Validation)

Outline

Information Extraction

- **Information Extraction** befasst sich mit der Extraktion von Semantik aus Texten.
- Im Allgemeinen ist dies nur sehr eingeschränkt möglich: Extraktion einfacher **Fakten**

Hier: Named Entities

- Eine "named entity" ist ein **eindeutiges, benamtes Objekt**.

Named Entities	**Keine Named Entities**
Donald J. Trump, McDonalds, Nile, 13.04.2017, 20$	sand, cat, company

- Oft werden *(siehe Beispiele)* Datums- oder Zahlenangaben dazugezählt.
- In der Regel werden Entitäten auch **Klassen** zugeordnet *(z.B. "person", "organization", "money", "location")*

Named Entity Recognition (NER): Beispiel Bild: [1]

Citing high fuel prices, [ORG **United Airlines**] said [TIME **Friday**] it has increased fares by [MONEY **$6**] per round trip on flights to some cities also served by lower-cost carriers. [ORG **American Airlines**], a unit of [ORG **AMR Corp.**], immediately matched the move, spokesman [PER **Tim Wagner**] said. [ORG **United**], a unit of [ORG **UAL Corp.**], said the increase took effect [TIME **Thursday**] and applies to most routes where it competes against discount carriers, such as [LOC **Chicago**] to [LOC **Dallas**] and [LOC **Denver**] to [LOC **San Francisco**].

Anwendungen

- **Verknüpfung** von Texten mit strukturierten Quellen *(z.B. Wikipedia)*
- **Sentiment-Analyse**: *Über wen* wird gesprochen?
- **Question Answering**: Detektion von *Anwort-Kandidaten*

NER: Problemstellung

1. **Klassifikation**: Gegeben einen Satz, weise jedem Term ein Label *(z.B. PERSON, ORGA, LOCATION, TIME,...)* zu.
   Sonderfall: O ("Other", gehört zu keiner Named Entity).

"[PERSON Washington] [O was] [O born] [O into] [O slavery]" vs.
"[O Blair] [O arrived] [O in] [LOC Washington] [O to] [O visit]."

2. **Segmentierung**: Wo sind die Grenzen zwischen Named Entities?

Als Angela Merkel Seehofer rügte...

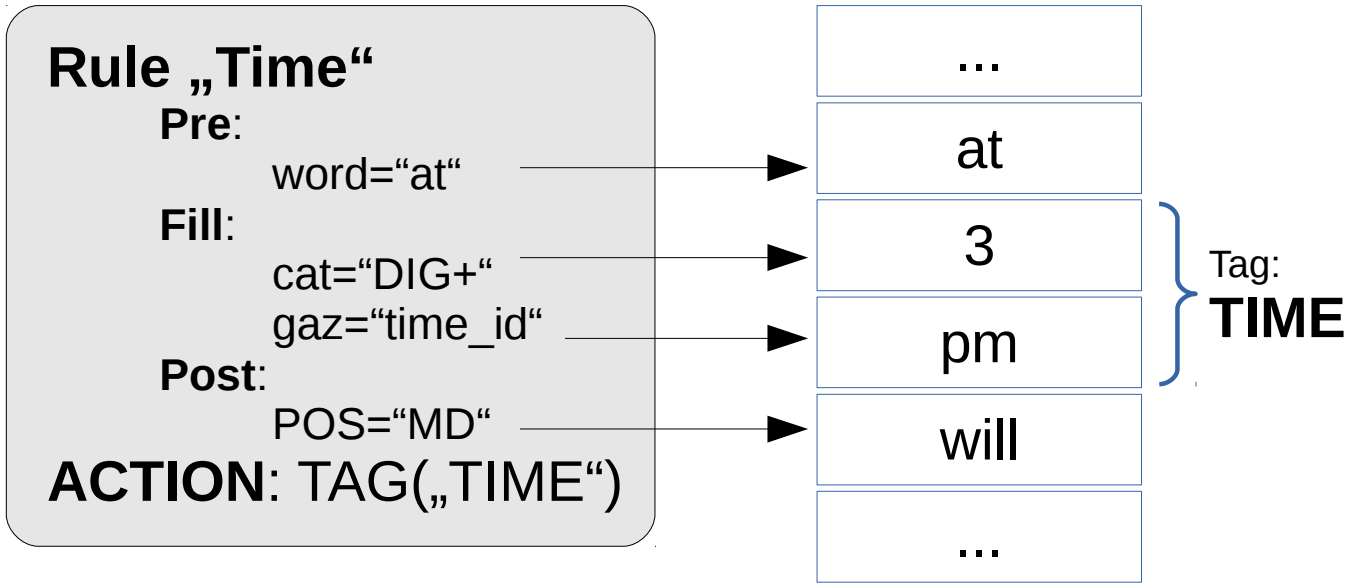
3. Clustering von Entitäten

Trump = the president = Agent Orange

NER: Problemstellung

Ansätze: Häufig eine Kombination aus...

- Regel-Systemen



- Matching von n-Grammen gegen Verzeichnisse **bekannter** Namen/Orte/Personen
- Machine Learning *(hier)*

NER mit Machine Learning

- Wir formulieren NER als sogenanntes **Sequence Labeling**: Gegeben eine Sequenz von Buchstaben $x_1, ..., x_n$, bestimme eine zugehörige Sequenz von Tags $t_1, ..., t_n$.
- Wir bestimmen den **Tag-Vokabular** geschickt so, dass die Labels sowohl den **Typ** als auch die **Grenzen** der Named Entities bestimmen!

3 Arten Labels ("BIO"-Labels)

1. **B-T**: Eine Named Entity mit Typ T **beginnt** hier *(B-PERSON, B-LOCATION, ...)*
2. **I-T**: Das Token gehört zu einer Named Entity mit Typ T, aber diese **beginnt hier nicht** *(I-PERSON, I-LOCATION, ...)*
3. **O**: Das Token gehört zu **keiner** Named Entity.

Americal	Airlines	,	a	company	in	New	York	,	...
B-ORG	I-ORG	O	O	O	O	B-LOC	I-LOC	O	...

NER mit ML: Merkmale

Welche Features sind interessant, um über die Zugehörigkeit eines Wortes zu einer Named Entity zu entscheiden?

- **Wortform** (X/x = Groß/Kleinbuchstabe, d = Ziffer)

  "We announced the TGX-42 today, which ..."
  → XXX-d (oder noch kürzer: X-d)

- **Präfixe/Suffixe** verschiedener Länge, z.B. $\mathbf{1}_{suffix(w)='ton'}$

  "He was in distress" vs. "He was in Fartington"

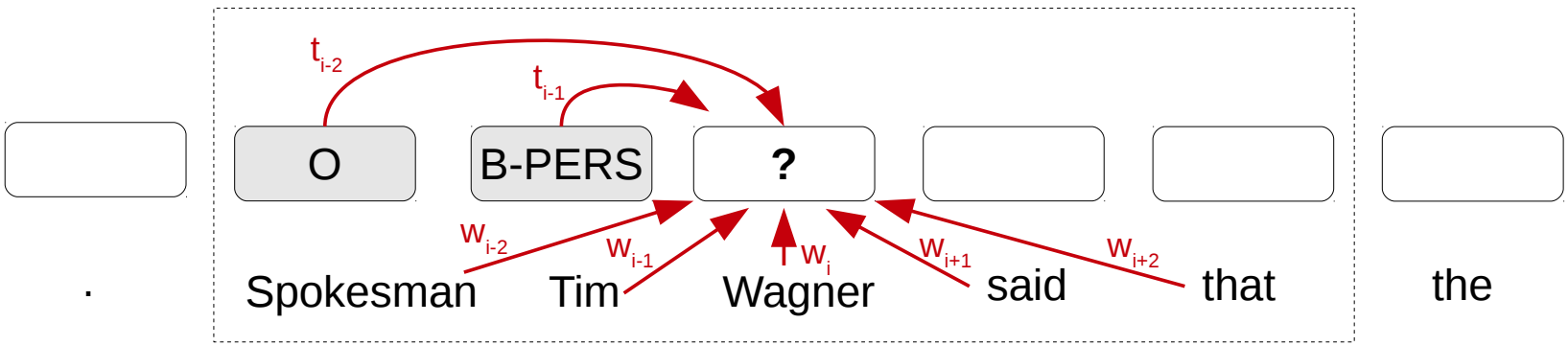
- **POS-Tags**, auch benachbarter Worte, z.B. $\mathbf{1}_{POS(w(-1))=PREPOSITION}$

  "...the CEO of Google..."

- **Identität** von Worten, z.B. $\mathbf{1}_{w(+1)='said'}$
- Vorhandensein in Verzeichnis bekannter Namen
- ...

NER: Klassifikator

- Um den Term x_i zu labeln, poolen wir Features aus einem **lokalen Fenster** $x_{i-w}, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_{i+w}$
- Wir berücksichtigen außerdem die **Tags/Labels** der **Vorgängerterme** $t_{i-w}, \ldots, t_{i-1}$



Inferenz/Klassifikation von Sequenzen

- **Verfahren 1**: Klassifiziere jeden Term von links nach rechts, treffe für jeden Term eine harte lokale Entscheidung *(Greedy-Decoding)*
- **Verfahren 2**: Globale Optimierung mit Viterbi-Coding [1] (Kapitel 10)

References I

[1] Daniel Jurafsky and James H. Martin.
Speech and Language Processing: An Introduction to Natural Language Processing (3rd Edition Draft Chapters).
2017.