Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Modelle, Daten, Lernprobleme

Tobias Scheffer

Überblick

- Arten von Lernproblemen:
  - Überwachtes Lernen (Klassifikation, Regression, ordinale Regression, Empfehlungen, Sequenzen und Strukturen)
  - Unüberwachtes Lernen
  - Reinforcement-Lernen (Exploration vs. Exploitation)
- Modelle
- Regularisierte empirische Risikominimierung
  - Verlustfunktionen
  - Regularisierer
- Evaluierung

Überwachtes Lernen: Grundbegriffe

- Instanz $\mathbf{x} \in X$
  - In der Statistik: unabhängige Variable
  - X kann z.B. Vektorraum über *Attribute* sein ($X = \mathbb{R}^m$)
  - Instanz ist in diesem Fall Belegung der Attribute.
  - $\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$ Merkmalsvektor
- Zielvariable: $y \in Y$
  - In der Statistik: abhängige Variable
- Ein Modell bildet Instanzen auf die Zielvariable ab.

$$\mathbf{x} \xrightarrow{\text{Modell}} y$$

Überwachtes Lernen: Klassifikation

- Eingabe: Instanz $\mathbf{x} \in X$.
  - z.B. Merkmalsvektor

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \dots \\ x_m \end{pmatrix}$$

- Ausgabe: Klasse $y \in Y$; endliche Menge Y.
  - Klasse wird auch als Zielattribut bezeichnet
  - y heißt auch (Klassen)Label

$$\mathbf{x} \xrightarrow{\text{Klassifikator}} y$$

Klassifikation: Beispiel

- Eingabe: $\mathbf{x} \in X$.

  X = Menge aller möglichen Kombinationen einer Menge von Wirkstoffem

Attribute	Instanz $\mathbf{x}$
Wirkstoff 1 enthalten?	$\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$
$\vdots$	
Wirkstoff 6 enthalten?	

Belegung der Attribute, Merkmalsvektor

Wirkstoff-kombination

- Ausgabe: $y \in Y = \{toxisch, ok\}$ /

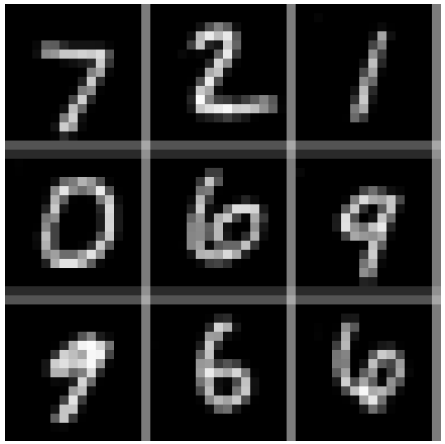
Klassifikator $\longrightarrow$

Klassifikation: Beispiel

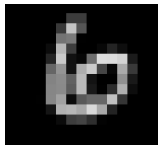
- Eingabe: Instanz $\mathbf{x} \in X$.

  X = Menge aller 16x16 Pixel Bitmaps

Attribute	Instanz $\mathbf{x}$
Grauwert Pixel 1	$\begin{pmatrix} 0.1 \\ 0.3 \\ 0.45 \\ \dots \\ 0.65 \\ 0.87 \end{pmatrix}$ 256 Pixelwerte
...	
Grauwert Pixel 256	



- Ausgabe: $y \in Y = \{0,1,2,3,4,5,6,7,8,9\}$: erkannte Ziffer



Klassifikator $\longrightarrow$ "6"

Klassifikation: Beispiel

- Eingabe: Instanz $\mathbf{x} \in X$.

X = Bag-of-Words-Repräsentation aller möglichen Email-Texte

Attribute	Instanz $\mathbf{x}$	
Wort 1 kommt vor?	0	Alternative
	1	Beneficiary
	0	Friend
…	…	…
	1	Sterling
Wort N kommt vor?	0	Zoo

N $\geq 1.000.000$

Email

Dear Beneficiary,

your Email address has been picked online in this years MICROSOFT CONSUMER AWARD as a Winner of One Hundred and Fifty Five Thousand Pounds Sterling…

- Ausgabe: $y \in Y = \{spam, ok\}$

Dear Beneficiary,

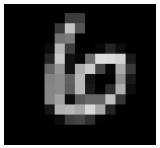
We are pleased to notify you that your Email address has been picked online in this second quarter's MICROSOFT CONSUMER AWARD (MCA) as a Winner of One Hundred and Fifty Five Thousand Pounds Sterling…

Klassifikator $\longrightarrow$ „Spam“

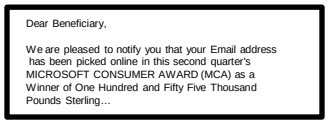
Klassifikation

- Klassifikator soll aus Trainingsdaten gelernt werden.

Klassifikator →



Klassifikator → "6"


Dear Beneficiary,

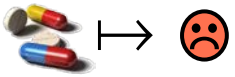
We are pleased to notify you that your Email address has been picked online in this second quarter's MICROSOFT CONSUMER AWARD (MCA) as a Winner of One Hundred and Fifty Five Thousand Pounds Sterling...

Klassifikator → „Spam“

Klassifikationslernen

- Eingabe Lernproblem: Trainingsdaten T_n .
  - $\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$
  - $\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$
- Trainingsdaten:

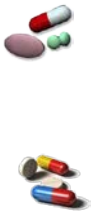
$$T_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$$

- Ausgabe: Modell
  -  $y_{\boldsymbol{\theta}} : X \to Y$
  - zum Beispiel

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ \mathbf{x}^{\mathrm{T}}\boldsymbol{\theta} \geq 0 \\ \text{☺}, & sonst \end{cases}$$

Linearer Klassifikator mit Parametervektor $\boldsymbol{\theta}$.

Klassifikationslernen

- Eingabe Lernproblem: Trainingsdaten T_n .
  - $\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$ 
  - $\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ ☺ ☹
- Trainingsdaten:

$$T_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$$

- Ausgabe: Modell
  - $\mapsto$ ☹

$$y_{\boldsymbol{\theta}} : X \to Y$$

- Modellklassen:
  - (Generalisierte) Lineare Modelle
  - Entscheidungsbäume
  - Ensemble-Klassifikatoren
  - …

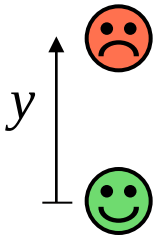
Überwachtes Lernen: Regression

- Eingabe: Instanz $\mathbf{x} \in X$.
  - z.B. Merkmalsvektor

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \dots \\ x_m \end{pmatrix}$$

Wie toxisch ist Kombination?

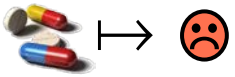
- Ausgabe: kontinuierlicher Wert, $y \in \mathbb{R}$
  - z.B *Toxizität*.



Regressionslernen

- Eingabe Lernproblem: Trainingsdaten T_n .
  - $\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$
  - $\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$
- Trainingsdaten:

$$T_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$$

- Ausgabe: Modell
  - 

$$y_{\boldsymbol{\theta}} : X \to Y$$

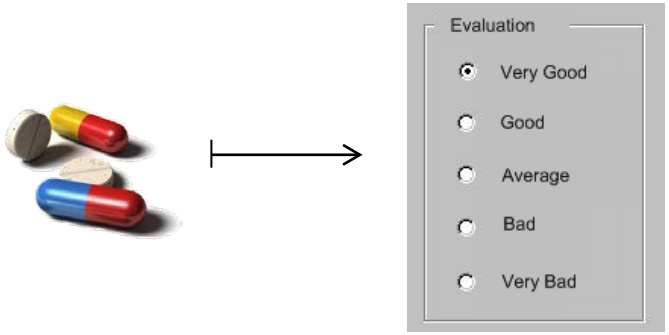
  - zum Beispiel

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\boldsymbol{\theta}$$

Generalisiertes lineares Modell mit Parametervektor $\boldsymbol{\theta}$.

Überwachtes Lernen: Ordinale Regression

- Eingabe: Instanz $\mathbf{x} \in X$.
- Ausgabe: diskreter Wert $y \in Y$, es existiert jedoch eine Ordnung auf den Elementen von Y.
- Große Abweichungen zwischen Vorhersage des Modells und echtem Wert sind schlechter als kleine.

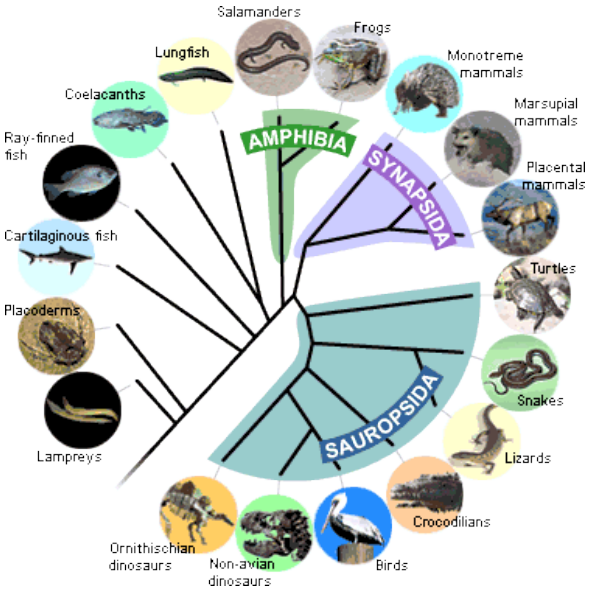


Zufrieden mit der Wirkung?

Überwachtes Lernen: Taxonomieklassifikation

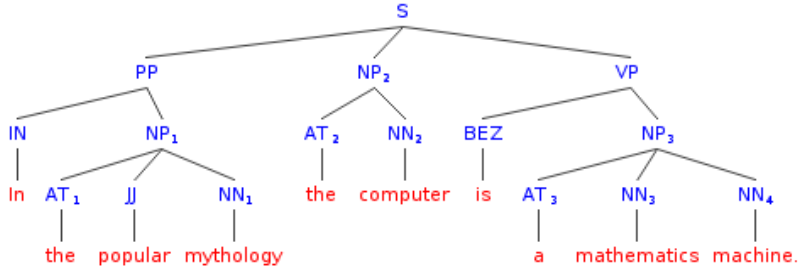
- Eingabe: Instanz $\mathbf{x} \in X$.
- Ausgabe: diskreter Wert $y \in Y$; es gibt eine baumförmige Ordnung auf den Elementen von Y.
- Vorhersage ist umso schlechter, je weiter vorhergesagter und tatsächlicher Knoten auseinanderliegen.

$\longmapsto$

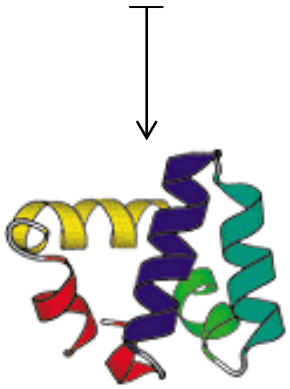


Überwachtes Lernen: Sequenz- und Strukturvorhersage

- Eingabe: Instanz $\mathbf{x} \in X$.
- Ausgabe: Sequenz, Baum oder Graph $y \in Y$.
- Beispielanwendungen:
  - Parsen natürlicher Sprache
  - Proteinfaltungen

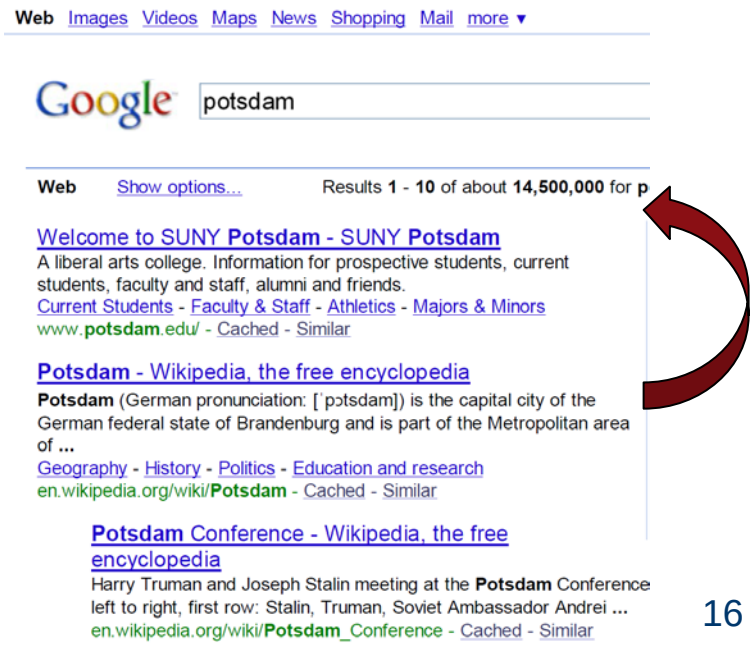


…AAGCTTGCACTGCCGT…



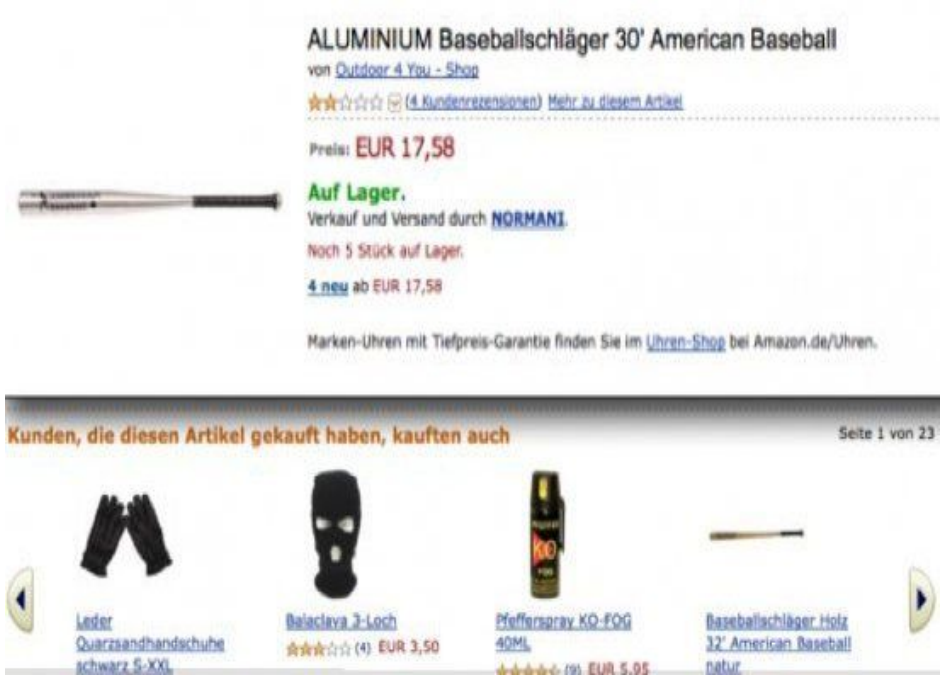
Überwachtes Lernen: Rankings

- Eingabe: Anfrage q, Liste von Items $I_1, \ldots, I_n$.
- Ausgabe: Sortierung der Items
- Trainingsdaten: Nutzer klickt nach Anfrage q auf I_j:
  - Item sollte höher gerankt werden als weiter oben stehende Items, die nicht angeklickt wurden.



Überwachtes Lernen: Empfehlungen

- Eingabe: Benutzer, Items, Kontextinformationen.
- Ausgabe: Wie sehr wird ein Vorschlag einem Benutzer gefallen?
- Trainingsdaten: Ratings, Käufe, Seitenaufrufe.

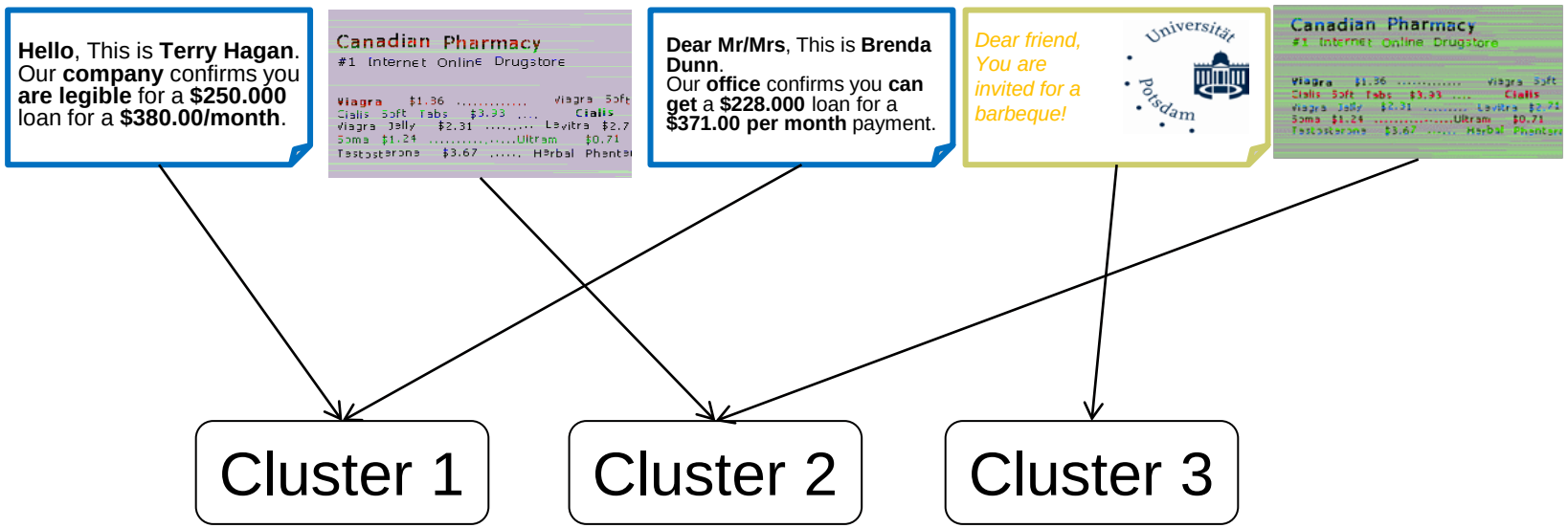


Unüberwachtes Lernen

- Trainingsdaten beim unüberwachten Lernen: Menge von Instanzen $\mathbf{x} \in X$.
- Zusätzliche Annahmen über Prozess der Entstehung der Daten, beispielsweise Unabhängigkeiten von Zufallsvariablen.
- Ziel ist das Aufdecken von Strukturen in den Daten:
  - Beispielsweise wahrscheinlichste Aufteilung in Cluster von Instanzen, die bestimmte, nicht direkt beobachtete Eigenschaften teilen.

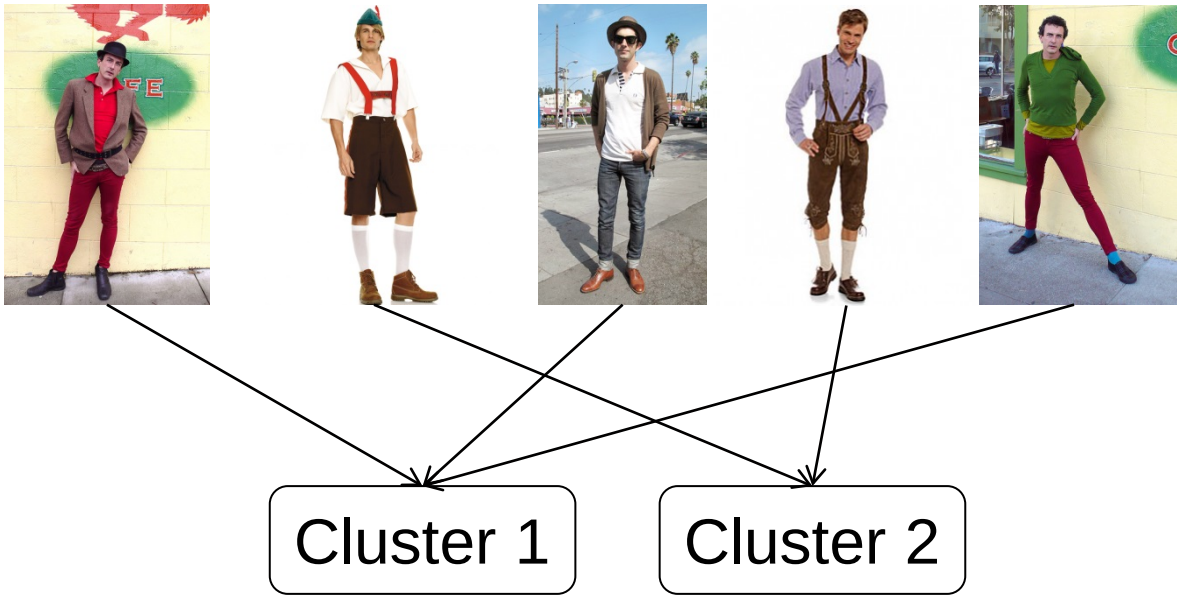
Clusteranalyse: Beispiel Emailkampagnen

- Eingabe: Strom von Emails.
- Ausgabe: Partitionierung in Teilmengen, die zur selben Kampagne gehören.



Clusteranalyse: Beispiel Marktsegmentierung

- Eingabe: Daten über Kundenverhalten.
- Ausgabe: Partitionierung in Cluster von Kunden, die ähnliche Produktpräferenzen haben.



Lernen von Steuerungen

- System, das über Steuerungsparameter verfügt.
- Nutzenfunktion beschreibt, welches Verhalten erwünscht ist.
- Änderung der Steuerungsparameter kann zeitlich verzögerte Auswirkungen haben.
- Lerner muss mit System experimentieren, um Modell für dessen Verhalten zu gewinnen (Exploration)
- Gleichzeitig soll das System möglichst in den besten Zuständen gehalten werden (Exploitation).

Lernen von Steuerungen: Beispiel

- Platzierung von Anzeigen.
- Um zu lernen, welche Anzeigen ein Benutzer anklickt, muss Lerner experimentieren.
- Wenn Lerner experimentiert statt bisher populärste Anzeigen zu verwenden, geht Umsatz verloren

Taxonomie von Lernproblemen

- Überwacht: Daten enthalten Werte für vorherzusagende Variable
  - Klassifikation: Kategorielle Variable
  - Regression: Kontinuierliche Variable
  - Ordinale Regression: Endliche, geordnete Menge
  - Rankings: Reihenfolge von Elementen
  - Strukturierte Vorhersage: Sequenz, Baum, Graph, …
  - Empfehlungen: Nutzer-Produkt-Matrix

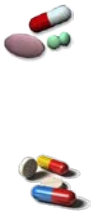
Taxonomie von Lernproblemen

- Unüberwacht: Strukturelle Eigenschaften der Trainingsdaten aufdecken
  - Clusteranalyse
  - Unüberwachtes Merkmalslernen: Finde Attribute, mit denn sich die Daten gut beschreiben lassen
- Steuerungen / Reinforcement-Lernen: Steuerung eines dynamischen Systems

Überblick

- Arten von Lernproblemen:
  - Überwachtes Lernen (Klassifikation, Regression, ordinale Regression, Empfehlungen, Sequenzen und Strukturen)
  - Unüberwachtes Lernen
  - Reinforcement-Lernen (Exploration vs. Exploitation)
- Modelle
- Regularisierte empirische Risikominimierung
  - Verlustfunktionen
  - Regularisierer
- Evaluierung

Klassifikationslernen

- Eingabe Lernproblem: Trainingsdaten T_n .
  - $\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{pmatrix}$ 
  - $\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ ☺ ☹
- Trainingsdaten:

$$T_n = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$$

- Ausgabe: Klassifikator
  - $\mapsto$ ☹

$$y_{\boldsymbol{\theta}} : X \to Y$$

- Wie kann Lernalgorithmus Modell (Klassifikator) aus Trainingsdaten lernen?
- Suchproblem im Raum aller Modelle

Modellraum, Parameterraum

- Modellraum, Parameterraum, Hypothesenraum Θ:
  - Klassifikator hat Parameter $\boldsymbol{\theta} \in \Theta$
  - Θ ist Menge der Modelle (Klassifikatoren), die Lernverfahren in Betracht zieht.
  - Modellraum ist einer der Freiheitsgrade beim maschinellen Lernen, viele Räume gebräuchlich.
  - Heisst auch *Language Bias*
- Beispiel:
  - Lineare Modelle

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn \sum_1^m x_j \boldsymbol{\theta}_j \geq \boldsymbol{\theta}_0 \\ \text{☺}, & sonst \end{cases}$$

Verlustfunktion, Optimierungskriterium

- Lernprobleme werden als Optimierungsaufgaben formuliert.
  - *Verlustfunktion* misst, wie gut Modell zu Trainingsdaten passt
  - *Regularisierungsfunktion* misst, ob das Modell nach unserem Vorwissen *wahrscheinlich* ist.
  - *Optimierungskriterium* ist Summe aus Verlust und Regularisierer.
  - Suche Minimum des Optimierungskriteriums

- Insgesamt wahrscheinlichstes Modell, gegeben Trainingsdaten und Vorwissen.

Verlustfunktion

- Verlustfunktion: Wie schlimm ist es, wenn Modell $y_{\boldsymbol{\theta}}(\mathbf{x}_i)$ vorhersagt obwohl der echte Wert der Zielvariable y_i ist?

$$\ell(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i)$$

- Mittlerer Verlust auf den gesamten Trainingsdaten T_n:
  - Empirisches Risiko $\hat{R}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i)$

- Beispiel: Binäres Klassifikationsproblem mit positiver Klasse (+1) und negativer Klasse (-1). False Positives und False Negatives gleich schlimm.
  - Zero-One Loss: $\ell_{0/1}(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) = \begin{cases} 0, & wenn\ y_{\boldsymbol{\theta}}(\mathbf{x}_i) = y_i \\ 1, & sonst \end{cases}$

Verlustfunktion

- Beispiel: diagnostische Klassifikationsprobleme, übersehene Erkrankungen (False Negatives) schlimmer als False Positives.
  - Kostenmatrix

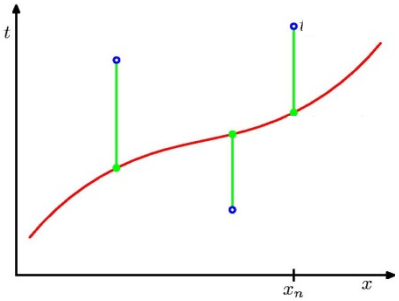
$$\ell_{c_{FP},c_{FN}}(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) = \begin{cases} y_{\boldsymbol{\theta}}(\mathbf{x}_i) = +1 \\ y_{\boldsymbol{\theta}}(\mathbf{x}_i) = -1 \end{cases}$$

	$y_i = +1$	$y_i = -1$
	0	c_{FP}
	c_{FN}	0

Verlustfunktion

- Beispiel Verlustfunktion Regression: Vorhersage möglichst dicht an echtem Wert des Zielattributes
  - Quadratischer Fehler

$$\ell_2(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) = (y_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i)^2$$



Verlustfunktion

- Wie schlimm ist es, wenn Modell y' vorhersagt obwohl der echte Wert der Zielvariable y ist?
  - Verlust $\ell(y', y)$

- Verlustfunktion ist aus der jeweiligen Anwendung heraus motiviert.

Suche nach Modell

- Suche nach Klassifikator für „Kombination toxisch“.

- Modellraum:

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn \sum_1^m x_j \boldsymbol{\theta}_j \geq \boldsymbol{\theta}_0 \\ \text{☺}, & sonst \end{cases}$$

Trainingsdaten

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	1	1	0	0	1	1	☹
$\mathbf{x}_2$	0	1	1	0	1	1	☹
$\mathbf{x}_3$	1	0	1	0	1	0	☺
$\mathbf{x}_4$	0	1	1	0	0	0	☺

- Ansatz: empirisches Risiko sollte minimal sein

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^n \ell_{0/1}(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i)$$

- Gibt es ein solches Modell? Gibt es mehrere?

Suche nach Modell

- Suche nach Klassifikator für „Kombination toxisch“.

- Modellraum:

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn \ \sum_1^m x_j \boldsymbol{\theta}_j \geq \boldsymbol{\theta}_0 \\ \text{☺}, & sonst \end{cases}$$

- Modelle mit Verlust von 0:
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹}, \ wenn \ x_6 \geq 1$
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹}, \ wenn \ x_2 + x_5 \geq 2$
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹}, \ wenn \ 2x_4 + x_6 \geq 1$
  - …

Trainingsdaten

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	1	1	0	0	1	1	☹
$\mathbf{x}_2$	0	1	1	0	1	1	☹
$\mathbf{x}_3$	1	0	1	0	1	0	☺
$\mathbf{x}_4$	0	1	1	0	0	0	☺

Suche nach Modell

- Suche nach Klassifikator für „Kombination toxisch“.

- Modellraum:

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn \sum_1^m x_j \boldsymbol{\theta}_j \geq \boldsymbol{\theta}_0 \\ \text{☺}, & sonst \end{cases}$$

- Modelle mit Verlust von 0:
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹},\ wenn\ x_6 \geq 1$
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹},\ wenn\ x_2 + x_5 \geq 2$
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹},\ wenn\ 2x_4 + x_6 \geq 1$
  - …

Trainingsdaten

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	1	1	0	0	1	1	☹
$\mathbf{x}_2$	0	1	1	0	1	1	☹
$\mathbf{x}_3$	1	0	1	0	1	0	☺
$\mathbf{x}_4$	0	1	1	0	0	0	☺

- Modelle mit einem empirischem Risiko von 0 bilden den *Versionenraum*.
- Versionenraum ist leer bei widersprüchlichen Daten.

Suche nach Modell

- Suche nach Klassifikator für „Kombination toxisch“.

- Modellraum:

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn \sum_1^m x_j \boldsymbol{\theta}_j \geq \boldsymbol{\theta}_0 \\ \text{☺}, & sonst \end{cases}$$

- Modelle mit Verlust von 0:
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹},\ wenn\ x_6 \geq 1$
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹},\ wenn\ x_2 + x_5 \geq 2$
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \text{☹},\ wenn\ 2x_4 + x_6 \geq 1$
  - …

Trainingsdaten

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	1	1	0	0	1	1	☹
$\mathbf{x}_2$	0	1	1	0	1	1	☹
$\mathbf{x}_3$	1	0	1	0	1	0	☺
$\mathbf{x}_4$	0	1	1	0	0	0	☺

- Modelle im Versionenraum unterscheiden sich in ihrer Vorhersage für einige Instanzen, die nicht in der Trainingsmenge vorkommen.
- Welches ist das Richtige?

Unsicherheit

- In der Praxis erreicht man niemals Gewissheit darüber, ein korrektes Modell gefunden zu haben.
- Daten können widersprüchlich sein (z.B. durch Messfehler)
- Viele unterschiedliche Modelle können einen niedrigen Verlust haben.
- Das korrekte Modell liegt vielleicht gar nicht im Modellraum.

- Lernen als *Optimierungsproblem*
  - Verlustfunktion: Grad der Konsistenz mit Trainingsdaten
  - Regularisierer: A-Priori-Wahrscheinlichkeit der Modelle

Regularisierer

- Regularisierer:
  - drückt Annahme darüber aus, ob Modell *a priori* wahrscheinlich ist.
  - Unabhängig von den Trainingsdaten.
  - Je höher der Regularisierungsterm für ein Modell, desto unwahrscheinlicher
- Häufig wird die Annahme ausgedrückt, dass wenige der Attribute für ein gutes Modell ausreichen.
  - Anzahl der Attribute, L_0-Regularisierung
  - Betrag der Attribut-Gewichtungen, L_1-Regularisierung
  - Quadrat der Attribut-Gewichtungen, L_2-Regularisierung.

Regularisierer

- Kandidaten:

$$(\theta_0\,\theta_1\,\theta_2\,\theta_3\,\theta_4\,\theta_5\,\theta_6)$$

$y_{\boldsymbol{\theta}}(\mathbf{x}) =$, $wenn\ x_6 \geq 1$ $\qquad \boldsymbol{\theta}_1 = (1\ 0\ 0\ 0\ 0\ 0\ 1)^{\mathrm{T}}$

$y_{\boldsymbol{\theta}}(\mathbf{x}) =$, $wenn\ x_2 + x_5 \geq 2$ $\qquad \boldsymbol{\theta}_2 = (2\ 0\ 1\ 0\ 0\ 1\ 0)^{\mathrm{T}}$

$y_{\boldsymbol{\theta}}(\mathbf{x}) =$, $wenn\ 2x_4 + x_6 \geq 1$ $\qquad \boldsymbol{\theta}_3 = (1\ 0\ 0\ 0\ 2\ 0\ 1)^{\mathrm{T}}$

- Regularisierer:

L_0-Regularisierung	L_1-Regularisierung	L_2-Regularisierung
$\Omega_0(\boldsymbol{\theta}_1) = 2$	$\Omega_1(\boldsymbol{\theta}_1) = 2$	$\Omega_2(\boldsymbol{\theta}_1) = 2$
$\Omega_0(\boldsymbol{\theta}_2) = 3$	$\Omega_1(\boldsymbol{\theta}_2) = 4$	$\Omega_2(\boldsymbol{\theta}_2) = 6$
$\Omega_0(\boldsymbol{\theta}_3) = 3$	$\Omega_1(\boldsymbol{\theta}_3) = 4$	$\Omega_2(\boldsymbol{\theta}_3) = 6$

Regularisiertes empirisches Risiko

- Trade-Off zwischen mittlerem Verlust und Regularisierer
  - $\frac{1}{n}\sum_{i=1}^{n}\ell(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) + \lambda\Omega(\boldsymbol{\theta})$
- Parameter λ steuert Trade-Off zwischen Verlust und Regularisierer.
- Viele Lernalgorithmen sind Spezialfälle der regularisierten empirischen Risikominimierung

Optimierungsproblem

- Rechtfertigung für Optimierungskriterium?
- Mehrere Rechtfertigungen und Herleitungen.
  - Wahrscheinlichstes Modell (*MAP-Modell*).
  - Niedrige obere Schranke für Fehler auf zukünftigen Daten abhängig von $|\boldsymbol{\theta}|$. (*SRM*).
  - Lernen ohne Regularisierung ist *ill-posed* Problem; keine eindeutige Lösung, oder Lösung hängt extrem stark von minimalen Änderungen in den Daten ab.

Regularisierte empirische Risikominimierung

- Suche nach Klassifikator für „Kombination toxisch“.

- Modellraum:
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn \sum_1^m x_j \boldsymbol{\theta}_j \geq \boldsymbol{\theta}_0 \\ \text{☺}, & sonst \end{cases}$

- Regularisiertes empirisches Risiko:
  - $\frac{1}{n} \sum_{i=1}^n \ell_{0/1}(y_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i) + 0.1\Omega_0(\boldsymbol{\theta})$

- Modell mit minimalem regularisiertem empirischem Risiko:
  - $y_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ x_2 \geq 1 \\ \text{☺}, & sonst \end{cases}$

Trainingsdaten

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	1	1	0	0	1	1	☹
$\mathbf{x}_2$	0	1	1	0	1	1	☹
$\mathbf{x}_3$	1	0	1	0	1	0	☺
$\mathbf{x}_4$	0	1	1	0	0	0	☺

Evaluierung von Modellen

- Wie gut wird ein Modell in Zukunft funktionieren?
- Zukünftige Instanzen werden entsprechend einer (unbekannten) Verteilung $p(\mathbf{x}, y)$ gezogen.
- Risiko: erwarteter Verlust unter $p(\mathbf{x}, y)$
  - $R(\boldsymbol{\theta}) = \sum_y \int \ell(y_{\boldsymbol{\theta}}(\mathbf{x}), y)\, p(\mathbf{x}, y)\, d\mathbf{x}$
- Ist das empirische Risiko auf den Trainingsdaten ein brauchbarer Schätzer für das Risiko?

Evaluierung von Modellen

- Wie gut wird ein Modell in Zukunft funktionieren?
- Zukünftige Instanzen werden entsprechend einer (unbekannten) Verteilung $p(\mathbf{x}, y)$ gezogen.
- Risiko: erwarteter Verlust unter $p(\mathbf{x}, y)$
- Ist das empirische Risiko auf den Trainingsdaten ein brauchbarer Schätzer für das Risiko?
  - Problem: Alle Modelle aus dem Version Space haben ein emp. Risiko von 0 auf Trainingsdaten.
  - Ein Klassifikator, der einfach nur die Trainingsdaten in einer Tabelle speichert hat ein empirisches Risiko von 0 auf Trainingsdaten.

Evaluierung von Modellen

- Wie gut wird ein Modell in Zukunft funktionieren?
- Zukünftige Instanzen werden entsprechend einer (unbekannten) Verteilung $p(\mathbf{x}, y)$ gezogen.
- Risiko: erwarteter Verlust unter $p(\mathbf{x}, y)$
- Empirisches Risiko auf den Trainingsdaten ist ein extrem optimistischer Schätzer für das Risiko.
- Risiko wird auf Daten geschätzt, die nicht zum Training verwendet wurden.
  - Training-and-Test.
  - N-fold Cross Validation.

Optimierungsproblem

- Einstellung von λ?
- Teile verfügbare Daten in Trainings- und Testdaten
- Iteriere über Werte für λ
  - Training auf Trainingsdaten, finde Modell
  - Bestimme auf Testdaten
- Wähle Wert λ mit minimalem Verlust
- Trainiere mit allen Daten

Daten, Modelle, Lernprobleme

- Überwachtes Lernen: Finde Modell, dass in Trainingsdaten beobachtete Zielvariable vorhersagt
- Verlustfunktion: Übereinstimmung zwischen Vorhersage des Modells und Wert der Zielvariable in den Trainingsdaten.
- Regularisierer: Übereinstimmung mit Vorwissen.
- Unüberwachtes Lernen: Keine Zielvariable, entdecke Struktur in den Daten, z.B. durch Einteilung der Instanzen in Cluster mit gemeinsamen Eigenschaften.
- Reinforcement-Lernen: Steuerung von Prozessen.