Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Maschinelles Lernen
Modelle, Version Spaces, Lernen

Christoph Sawade/Niels Landwehr
Silvia Makowski
Tobias Scheffer

Überblick

- Problemstellungen: Klassifikation und Regression
- Modelle und Hypothesenraum
- Verlustfunktionen und Regularisierer
- Unsicherheit, Wahrscheinlichkeiten, Bayes‘sche Regel

Überblick

- Problemstellungen: Klassifikation und Regression
- Modelle und Hypothesenraum
- Verlustfunktionen und Regularisierer
- Unsicherheit, Wahrscheinlichkeiten, Bayes‘sche Regel

Problemstellung Klassifikation

- Eingabe: Instanz (Objekt) $\mathbf{x} \in X$.
  - Objekte oft durch Vektor von *Attributen* repräsentiert ($X = \mathbb{R}^m$)
  - Instanz ist Belegung der Attribute.
  - $\mathbf{x} = \begin{pmatrix} x_1 \\ \dots \\ x_m \end{pmatrix}$ Merkmalsvektor

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

$\mathbf{x} \longrightarrow$ Klassifikator $\longrightarrow y$

Klassifikation: Beispiel

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

  X = Menge aller möglichen Kombinationen einer Menge von Medikamenten

  Attribute — Instanz $\mathbf{x}$

  Medikament 1 enthalten?

  $\vdots$

  Medikament 6 enthalten?

  $$\begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

  Belegung der Attribute, Merkmalsvektor

  Medikamenten-kombination

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

 → Klassifikator → 

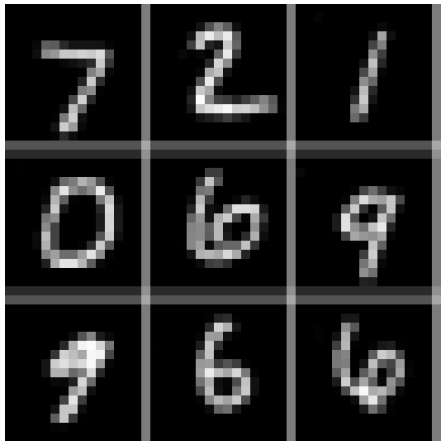
Klassifikation: Beispiel

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

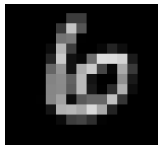
  X = Menge aller 16x16 Pixel Bitmaps

Attribute	Instanz $\mathbf{x}$
Grauwert Pixel 1	0.1
	0.3
	0.45
...	...
	0.65
Grauwert Pixel 256	0.87

256 Pixelwerte



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



→ Klassifikator → "6"

Klassifikation: Beispiel

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

  X = Menge 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

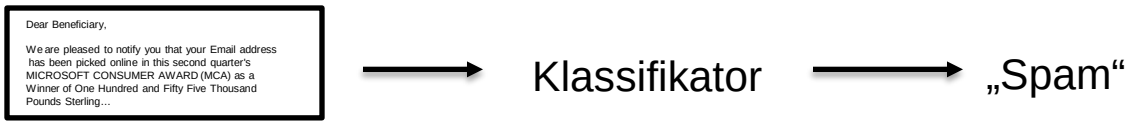
N ≈ 100000

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

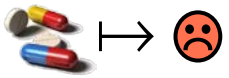


Problemstellung Klassifikationslernen

- Idee: Klassifikator aus Daten lernen
- Eingabe Lernproblem: Trainingsdaten.
  - $L = \langle (\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N) \rangle$
  - $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ ... \\ x_{im} \end{pmatrix}$

- Ausgabe: Klassifikator (auch als Modell bezeichnet).

$$f : X \rightarrow Y$$



$$f(\mathbf{x}) = \begin{cases} \text{☹}, & \textit{wenn } x_1 = 1 \wedge x_3 = 0 \wedge x_6 = 1 \\ \text{☺}, & \textit{sonst} \end{cases}$$

Problemstellung Klassifikationslernen

- Idee: Klassifikator aus Daten lernen
- Eingabe Lernproblem: Trainingsdaten.
  - $L = \langle (\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N) \rangle$
  - $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ ... \\ x_{im} \end{pmatrix}$
  - $\mathbf{x}_i$ Objektrepräsentation
  - y_i Klassenlabel

$$\left(\begin{pmatrix}1\\0\\0\\1\\0\\1\end{pmatrix}, ok\right) \quad \left(\begin{pmatrix}0\\1\\1\\0\\1\\0\end{pmatrix}, toxisch\right) \quad \left(\begin{pmatrix}1\\0\\0\\0\\1\\1\end{pmatrix}, ok\right)$$

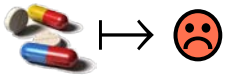
$$(\mathbf{x}_1, y_1) \quad (\mathbf{x}_2, y_2) \quad (\mathbf{x}_3, y_3)$$

Problemstellung Klassifikationslernen

- Eingabe Lernproblem: Trainingsdaten.
  - $L = \langle (\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N) \rangle$
  - $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ ... \\ x_{im} \end{pmatrix}$

- Ausgabe: Klassifikator (auch als Modell bezeichnet).

$$f : X \rightarrow Y$$



$$f(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ x_1 = 1 \wedge x_3 = 0 \wedge x_6 = 1 \\ \text{☺}, & sonst \end{cases}$$

$$f_{\mathbf{w}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ \mathbf{w}^{\mathrm{T}}\mathbf{x} + b \geq 0 \\ \text{☺}, & sonst \end{cases}$$

Linearer Klassifikator mit Parametervektor **w**.

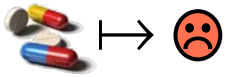
$$\mathbf{w}^{\mathrm{T}}\mathbf{x} = \sum_{i=1}^{m} w_i x_i$$

Problemstellung Klassifikationslernen

- Eingabe Lernproblem: Trainingsdaten.
  - $L = \langle (\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N) \rangle$
  - $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ ... \\ x_{im} \end{pmatrix}$

- Ausgabe: Klassifikator (auch als Modell bezeichnet).

$$f : X \rightarrow Y$$



Verschiedene Klassen von Klassifikatoren

- Entscheidungsbäume.
- Generalisierte lineare Modelle (Kernel).
-…
-Betrachtete Klassifikatoren wesentliches Unterscheidungsmerkmal zwischen Verfahren des ML

Problemstellung Klassifikationslernen

- Eingabe Lernproblem: Trainingsdaten.
  - $L = \langle (\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N) \rangle$
  - $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ ... \\ x_{im} \end{pmatrix}$

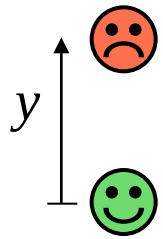
- Alternative Schreibweise:
  - Trainingsinstanzen: Matrix $\mathbf{X} = (\mathbf{x}_1 \quad ... \quad \mathbf{x}_N) = \begin{pmatrix} x_{11} & x_{N1} \\ \vdots & \ddots & \vdots \\ x_{1m} & x_{Nm} \end{pmatrix}$
  - Trainingslabels: Vektor $\mathbf{y} = \begin{pmatrix} y_1 \\ ... \\ y_N \end{pmatrix}$

Problemstellung: Regression

- Eingabe: Instanz (Objekt) $\mathbf{x} \in X$.
  - Objekte oft durch Attribut-Vektoren repräsentiert.
  - Instanz ist Belegung der Attribute.
  - $\mathbf{x} = \begin{pmatrix} x_1 \\ \dots \\ x_m \end{pmatrix}$ Merkmalsvektor

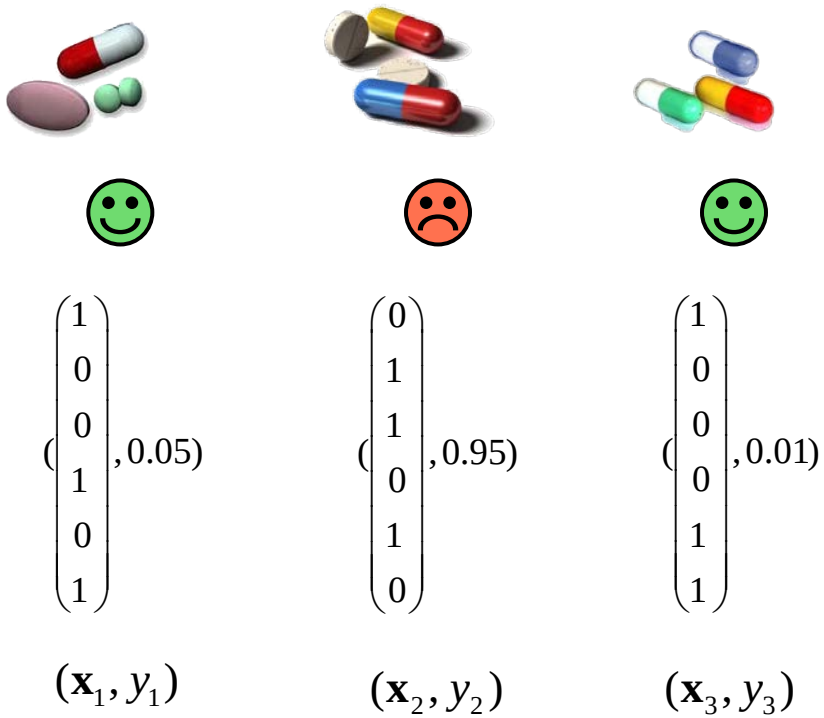
Wie toxisch ist Kombination?

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



Regressionslernen

- Eingabe: Trainingsdaten.
  - $L = \langle (\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N) \rangle$
  - $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ ... \\ x_{im} \end{pmatrix}$
  - $y_i \in \mathbb{R}$



- Ausgabe: Modell, Regressionsmodell.
  - $f : X \to \mathbb{R}$
  - Z.B. $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$

Andere Lernprobleme

- Ordinale Regression.
- Präferenzlernen.
- Taxonomie-Klassifikation.
- Klassifikation und Regression mit strukturierten Ausgaberäumen.
- Kollaborative Vorhersage.
- …

Andere Lernprobleme

- Ordinale Regression.
- Präfer[illegible]lernen
- [illegible]
-
-
-

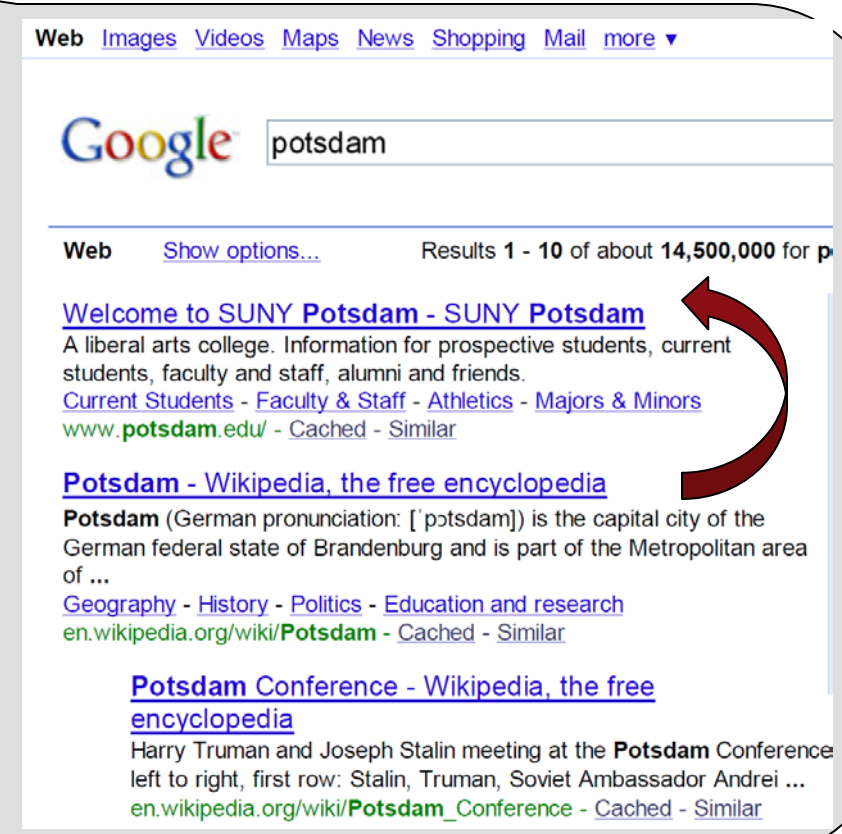
Mischung aus Klassifikation und Regression

- Endliche, diskrete Labels
- Ordnung

Andere Lernprobleme

- Ordinale Regression.
- Präferenzlernen.
- Taxon… Klassifikation.
- …
- …
- …

- Keine direkten Klassen beobachtet, sondern nur Präferenzen
- z.B Reihenfolge von Suchresultaten aus Clickstreams lernen

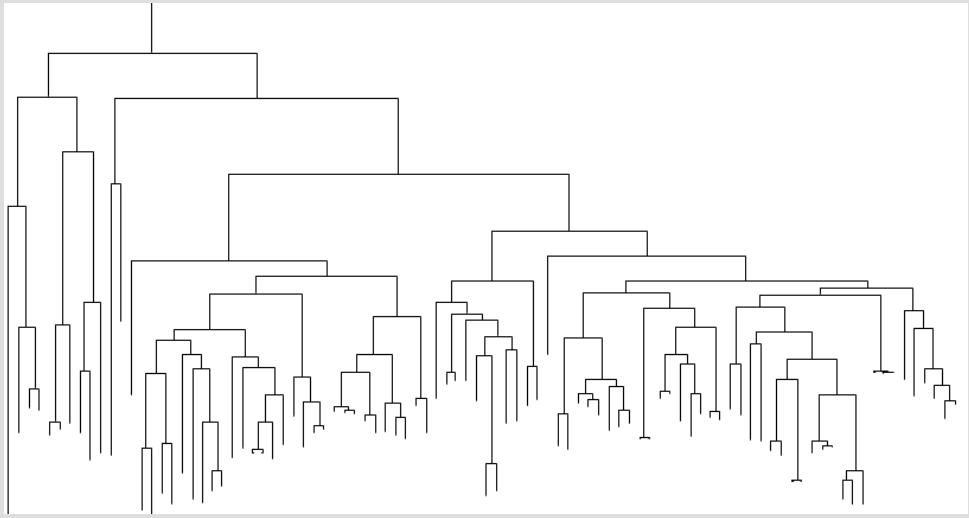


Andere Lernprobleme

- Ordinale Regression.
- Präferenzlernen.
- Taxonomie-Klassifikation.
- Klassifikation und Regression mit strukturierten
- Ein Objekt hat mehrere
- Klassenlabels

Hierarchie von Klassen

Panther ist…
->Tier
->Säugetier
->Katze
->Panther



Andere Lernprobleme

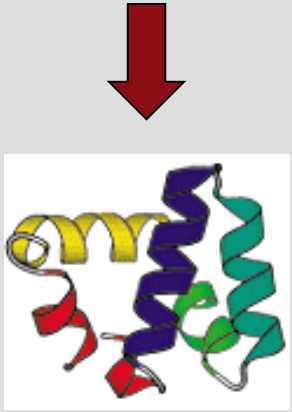
- Ordinale Regression.
- Präferenzlernen.
- Taxonomie-Klassifikation.
- Klassifikation und Regression mit strukturierten Ausgaberäumen.
- Kollabo[illegible]
- [illegible]

Eingabe X und Ausgabe Y
strukturierte Räume

Beispiel:
Eingabe DNA
Ausgabe Proteinfaltung

Klassenlabel 3D Struktur

…AAGCTTGCACTGCCGT…



Ausnutzen von Relationen zwischen Objekten

Beispiel: Produktempfehlungen

Vorhersage interessanter Produkte

Was hat der Nutzer/haben ähnliche Nutzer vorher gekauft?

- Kollaborative Vorhersage.
- …

Überblick

- Lernprobleme: Klassifikation und Regression
- Modelle und Hypothesenraum
- Verlustfunktionen und Regularisierer
- Unsicherheit, Wahrscheinlichkeiten, Bayes‘sche Regel

Klassifikationslernen

- Eingabe: Trainingsdaten.
  - $L = \langle (\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N) \rangle$
  - $\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ ... \\ x_{im} \end{pmatrix}$

- Ausgabe: Klassifikator.
  - $f : X \rightarrow Y$

$$f(\mathbf{x}) = \begin{cases} \text{☹}, & \text{wenn } x_1 = 1 \wedge x_3 = 0 \wedge x_6 = 1 \\ \text{☺}, & \text{sonst} \end{cases}$$

- Wie Klassifikator lernen aus Trainingsdaten?
  - Ansatz: Klassifikator, der Trainingsdaten (Beobachtungen) erklärt
  - Suchproblem im Raum aller (betrachteten) Klassifikatoren

Hypothesenraum

- Hypothesenraum, Modellraum H:
  - Menge der Klassifikationsmodelle, die Lernverfahren in Betracht zieht.
  - Hypothesenraum ist einer der Freiheitsgrade beim maschinellen Lernen, viele Räume gebräuchlich.
  - Hypothesenraum heisst auch *Language Bias*
- Beispiel:
  - Alle möglichen Konjunktionen von Bedingungen

$$f(\mathbf{x}) = \begin{cases} \text{☹}, & wenn \bigwedge_{j \in J} x_j = v_j \\ \text{☺}, & sonst \end{cases} \qquad J \subseteq \{1,...,m\}, \quad v_j \in \{0,1\}$$

  - Wie groß ist Hypothesenraum (*m* binäre Attribute)?

Suche nach Hypothese

- Suche nach Klassifikator für „Kombination toxisch“.
- Hypothesenraum:

$$f(\mathbf{x}) = \begin{cases} ☹, & \textit{wenn } \bigwedge_{j \in J} x_j = v_j \\ ☺, & \textit{sonst} \end{cases}$$

Trainingsdaten

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	0	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: Hypothese sollte konsistent sein mit Trainingsdaten

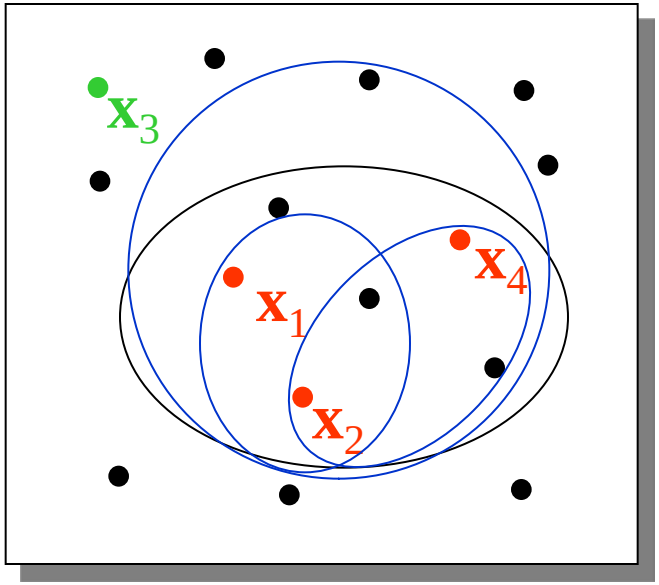
$$\forall i : f(\mathbf{x}_i) = y_i$$

- Identifizieren aller solchen Hypothesen?
- Nutze Struktur auf dem Hypothesenraum (generell/speziell)

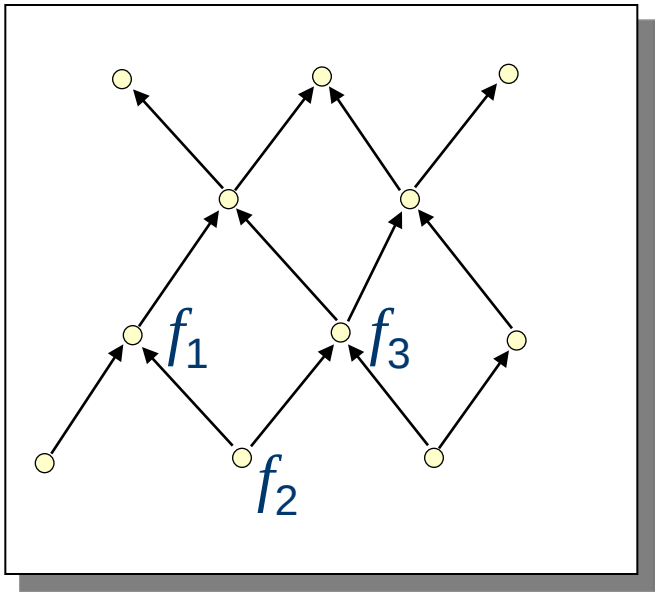
„Genereller-Als“-Ordnung

$$f_g \geq_g f_s \quad gdw. \quad f_s(\mathbf{x}) = ☹ \Rightarrow f_g(\mathbf{x}) = ☹ \; \forall \mathbf{x} \in X$$

Grundmenge X



Hypothesen H



spezifisch

Immer ☺

generell

Immer ☹

f_1= ☹, wenn x_2=1, x_6=1 f_2= ☹, wenn x_2=1 f_3= ☹, wenn x_2=1, x_3=1

$$f_2 \geq_g f_1 \quad f_2 \geq_g f_3 \; aber \; nicht \; f_1 \; {}^{\leq_g}\!/_{\geq_g} \; f_3$$

Version Space

- Menge aller Hypothesen, die mit den Trainingsdaten konsistent sind, nennen wir den Version Space:

$$VS_{H,L} = \{ f \in H | \forall (\mathbf{x}_i, y_i) \in L : f(\mathbf{x}_i) = y_i \}$$

- Version Space begrenzt durch generellste/speziellste Hypothesen, die Daten erklären

$$G = \{ f \in VS_{H,L} \mid \neg \exists f' \in VS_{H,L} : f' >_g f \}$$

Generellste konsistente Hypothesen

$$S = \{ f \in VS_{H,L} \mid \neg \exists f' \in VS_{H,L} : f >_g f' \}$$

Speziellste konsistente Hypothesen

Version Space: alles „zwischen“ *G* und *S* (keine unendlichen Ketten)

$$VS_{H,L} = \{ f \in H \mid \exists f_g \in G, f_s \in S : f_g \geq_g f \geq_g f_s \}$$

Version Space: Beobachtungen

$$VS_{H,L} = \{ f \in H | \forall (\mathbf{x}_i, y_i) \in L : f(\mathbf{x}_i) = y_i \}$$

- Version Space wird kleiner, je mehr Daten vorhanden
- Version Space leer: Trainingsmenge widersprüchlich (es existiert keine Hypothese in H, die Daten erklärt)
- Version Space einelementig:
  - Richtiges Modell gefunden,
  - Oder richtiges Modell ist nicht im Hypothesenraum.
- Mehrere Elemente im Version Space:
  - Noch nicht fertig.

Version Space: Brute Force Konstruktion

- Initialisiere V auf Menge aller Hypothesen.
- Für alle Trainingsbeispiele $(\mathbf{x}_i, y_i)$:
  - Lösche alle Hypothesen f aus V, die mit $\mathbf{x}_i$ inkonsistent sind, also $f(\mathbf{x}_i) \neq y_i$.
- V ist jetzt der Version Space
- Bessere Verfahren unter Benutzung von *G* und *S* (keine Details)

Beispiel: Version Space

- H: $f(\mathbf{x}) = \begin{cases} \text{☹}, & \textit{wenn } \bigwedge_{j \in J} x_j = v_j \\ \text{☺}, & \textit{sonst} \end{cases}$

- Welche Hypothesen sind im Version Space?

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	0	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	☹

$$L = \langle (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), (\mathbf{x}_4, y_4) \rangle$$

Beispiel: Version Space

- H: $f(\mathbf{x}) = \begin{cases} \text{☹}, & \text{wenn } \bigwedge_{j \in J} x_j = v_j \\ \text{☺}, & \text{sonst} \end{cases}$

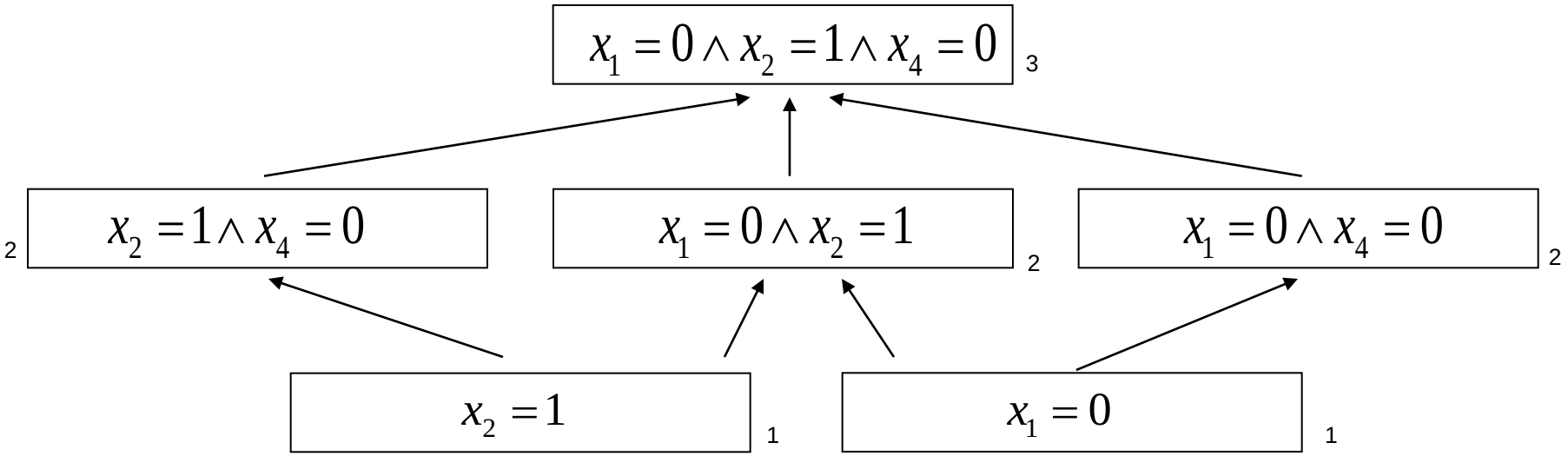
- Welche Hypothesen sind im Version Space?

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	0	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	☹

$$L = \langle (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), (\mathbf{x}_4, y_4) \rangle$$



Version Space

- H: $f(\mathbf{x}) = \begin{cases} \text{☹}, & \text{wenn } \bigwedge_j x_j = v_j \\ \text{☺}, & \text{sonst} \end{cases}$

- Welche Hypothesen sind im Version Space?

Medikamente in der Kombination

Beispiel-Kombinationen

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	0	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	☹

- Problem:
  - Alle Elemente des Version Space erklären die Daten gleichermaßen gut.
  - Version Space nicht robust bei fehlerhaften Daten

Unsicherheit

- In der Praxis erreicht man niemals Gewissheit darüber, ein korrektes Modell gefunden zu haben.
- Version Space-Ansatz problematisch
  - Der Hypothesenraum ist meist unendlich groß.
  - Der Version Space ist dann meist auch unendlich groß, oder leer.

Alternative/zusätzliche Konzepte

- Lernen als *Optimierungsproblem*
  - Verlustfunktionen: Grad der Konsistenz mit Trainingsdaten
  - A-priori Verteilung über Modelle, Regularisierer

Überblick

- Lernprobleme: Klassifikation und Regression
- Modelle und Hypothesenraum
- Verlustfunktionen und Regularisierer
- Unsicherheit, Wahrscheinlichkeiten, Bayes‘sche Regel

Verlustfunktion, Optimierungskriterium

- Alternative zu Version Spaces: Lernprobleme werden als Optimierungsprobleme 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

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

$$l(f(\mathbf{x}_i), y_i)$$

- Verlust auf den ganzen Trainingsdaten *L*:

$$\sum_{i=1}^{N} l(f(\mathbf{x}_i), y_i)$$

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

Verlustfunktion

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

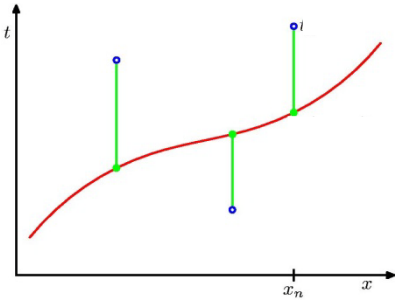
$$l(f(\mathbf{x}_i), y_i) = \begin{cases} f(\mathbf{x}_i) = +1 \\ f(\mathbf{x}_i) = -1 \end{cases}$$

	$y_i = +1$	$y_i = -1$
$f(\mathbf{x}_i) = +1$	0	c_{FP}
$f(\mathbf{x}_i) = -1$	c_{FN}	0

Verlustfunktion

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

$$l(f(\mathbf{x}_i), y_i) = (f(\mathbf{x}_i) - y_i)^2$$



Verlustfunktion

- Wie schlimm ist es, wenn Modell $f(\mathbf{x}_i)$ vorhersagt obwohl der echte Wert der Zielvariable y_i ist?
  - Verlust $l(f(\mathbf{x}_i), y_i)$.

- Verlustfunktion ist aus der jeweiligen Anwendung heraus motiviert.

Regularisierer

- Verlustfunktion drückt aus, wie gut Modell zu Daten passt
- 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: Beispiel

- Hypothesenraum: Konjunktion von Bedingungen
  - $f(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ x_1 = 1 \wedge x_3 = 1 \wedge x_7 = 1 \\ \text{☺}, & sonst \end{cases}$
- Lineares Modell: Lässt sich schreiben als
  - $f(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ x_1 + x_3 + x_7 \geq 3 \\ \text{☺}, & sonst \end{cases}$
- Allgemein: äquivalente Darstellung ist
  - $f_{\mathbf{w}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ \sum_{j=1}^{m} w_j x_j \geq b \\ \text{☺}, & sonst \end{cases}$

    $= \begin{cases} \text{☹}, & wenn\ \mathbf{w}^{\mathrm{T}}\mathbf{x} \geq b \\ \text{☺}, & sonst \end{cases}$

    w: Modellparameter

$w_j \in \{-1, 0, +1\}$

$w_j \in \{-1, +1\}$ falls Attribut in logischer Bedingung vorkommt

Regularisierer: Beispiel

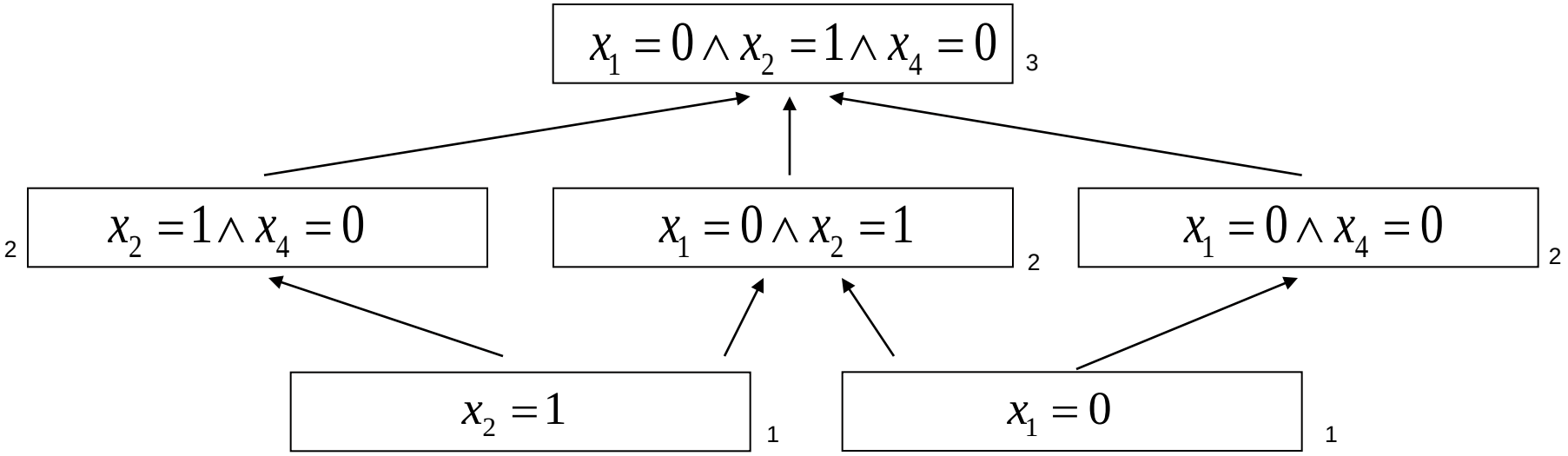
- Linearer Klassifikator
  - $f_{\mathbf{w}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ \mathbf{w}^{\mathrm{T}}\mathbf{x} \geq b \\ \text{☺}, & sonst \end{cases}$
- L_2-Regularisierung:
  - $\lambda \,|\, \mathbf{w} \,|^2 \qquad |\, \mathbf{w} \,|^2 = \sum_i w_i^2$
  - Addiert λ für jedes von null verschiedene Gewicht.
- Optimierungskriterium: Verlust+Regularisierer
  - $\hat{R}(\mathbf{w}, L) = \sum_i l(\, f_{\mathbf{w}}(\mathbf{x}_i), y_i) + \lambda \,|\, \mathbf{w} \,|^2$
  - Parameter λ steuert Stärke des Regularisierers

- Durch den Regularisierer implementierte Präferenz des Lerners wird auch *Inductive Bias* genannt.

Optimierungsproblem: Beispiel

- $\hat{R}(\mathbf{w}, L) = \sum_i l(f_{\mathbf{w}}(\mathbf{x}_i), y_i) + \lambda |\mathbf{w}|^2$
- Beste Hypothese für $\lambda = 0.1$?

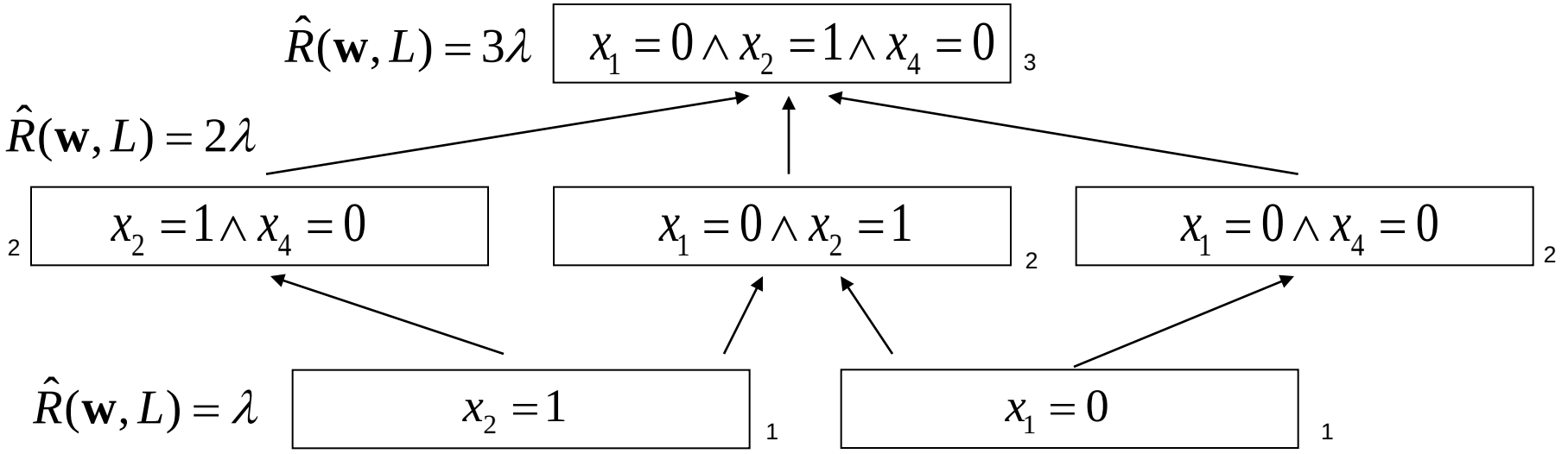
	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	0	1	0	0	1	1	☹
$\mathbf{x}_2$	0	1	1	0	1	1	☹
$\mathbf{x}_3$	1	0	1	0	1	0	☺
$\mathbf{x}_4$	0	1	1	0	0	0	☹



Optimierungsproblem: Beispiel

- $\hat{R}(\mathbf{w}, L) = \sum_i l(f_{\mathbf{w}}(\mathbf{x}_i), y_i) + \lambda \,|\, \mathbf{w} \,|^2$
- Beste Hypothese für $\lambda = 0.1$?

	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	0	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	☹



Optimierungsproblem

- Einstellung von λ?
- Rechtfertigung für Optimierungskriterium?
- Mehrere Rechtfertigungen und Herleitungen.
  - **Wahrscheinlichste Hypothese (*MAP-Hypothese*).**
  - Hypothese, die Daten am stärksten komprimiert (*Minimum Description Length*).
  - Niedrige obere Schranke für Fehler auf zukünftigen Daten abhängig von $|\mathbf{w}|$. (*SRM*).
- Lernen ohne Regularisierung ist *ill-posed* Problem; keine eindeutige Lösung, oder Lösung hängt extrem stark von minimalen Änderungen in den Daten ab.

Überblick

- Lernprobleme: Klassifikation und Regression
- Modelle und Hypothesenraum
- Verlustfunktionen und Regularisierer
- Unsicherheit, Wahrscheinlichkeiten, Bayes‘sche Regel