Universität Potsdam
Institut für Informatik
Lehrstuhl Maschinelles Lernen

Maschinelles Lernen
Modelle, Version Spaces, Lernen

Christoph Sawade/Niels Landwehr
Jules Rasetaharison
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

Klassifikation

- Eingabe: Instanz (Objekt) $\mathbf{x} \in X$.
  - Objekte oft durch Vektor von *Attributen* repräsentiert
  - 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

  Medikament 1 enthalten?

  ⋮

  Medikament 6 enthalten?

  Instanz $\mathbf{x}$

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

  Belegung der Attribute, Merkmalsvektor

  Medikamenten-kombination

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

 → Klassifikator → 

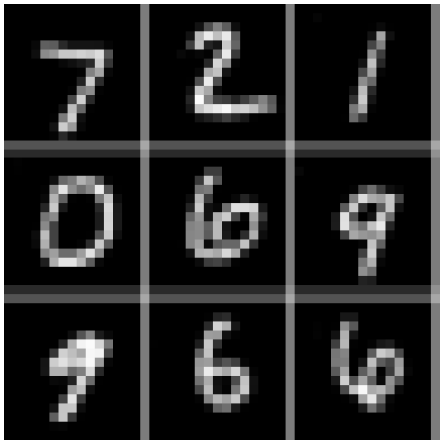
Klassifikation: Beispiel

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

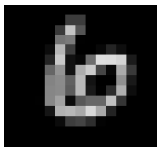
  X = Menge aller 16x16 Pixel Bitmaps

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

Wort 1 kommt vor?

…

Wort N kommt vor?

$N \approx 100000$

Instanz $\mathbf{x}$

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

Address
Beneficiary
Friend
…
Sterling
Science

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 → „Spam“

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

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

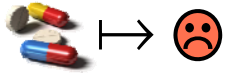
$$(\mathbf{x}_1, y_1) \qquad (\mathbf{x}_2, y_2) \qquad (\mathbf{x}_3, y_3)$$

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).

$f : X \rightarrow Y$



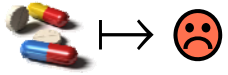
$$f(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ x_1 = 1 \wedge x_3 = 0 \wedge x_6 = 1 \\ \text{☺}, & sonst \end{cases}$$

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).

$f : X \to Y$



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

Linearer Klassifikator mit Parametervektor **w**.

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

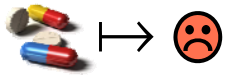
$$\mathbf{w}^T\mathbf{x} = \sum_{i=1}^{m} w_i x_i$$

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

Klassifikationslernen

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

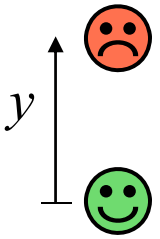
- Alternative Schreibweise:
  - Trainingsinstanzen: Matrix $\mathbf{X} = (\mathbf{x}_1 \quad ... \quad \mathbf{x}_N) = \begin{pmatrix} x_{11} & x_{N1} \\ \vdots & \ddots & \vdots \\ x_{1m} & x_{Nm} \end{pmatrix}$
  - Trainingslabels: Vektor $\mathbf{y} = \begin{pmatrix} y_1 \\ ... \\ y_N \end{pmatrix}$

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

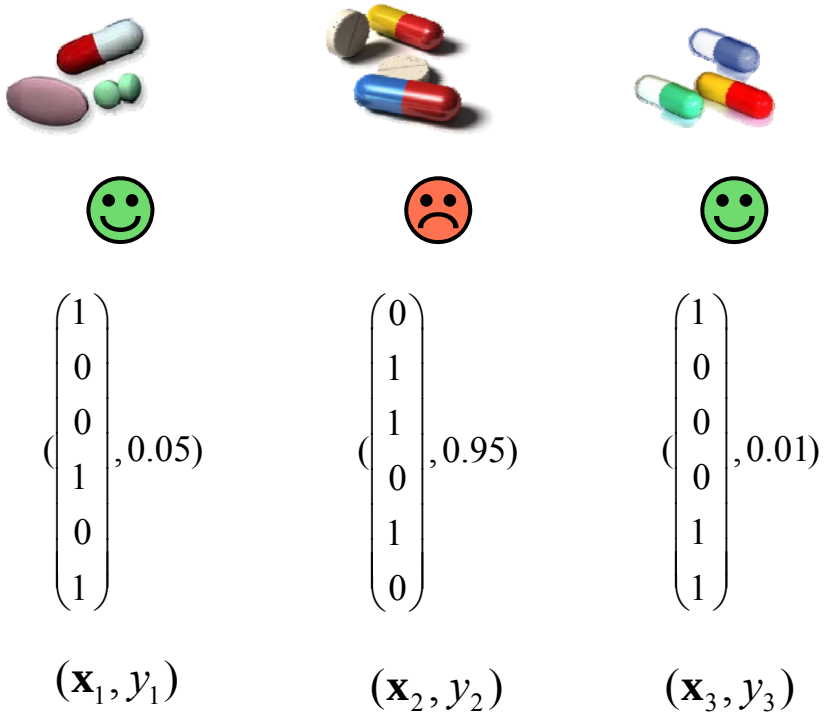
Wie toxisch ist Kombination?

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



Regressionslernen

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

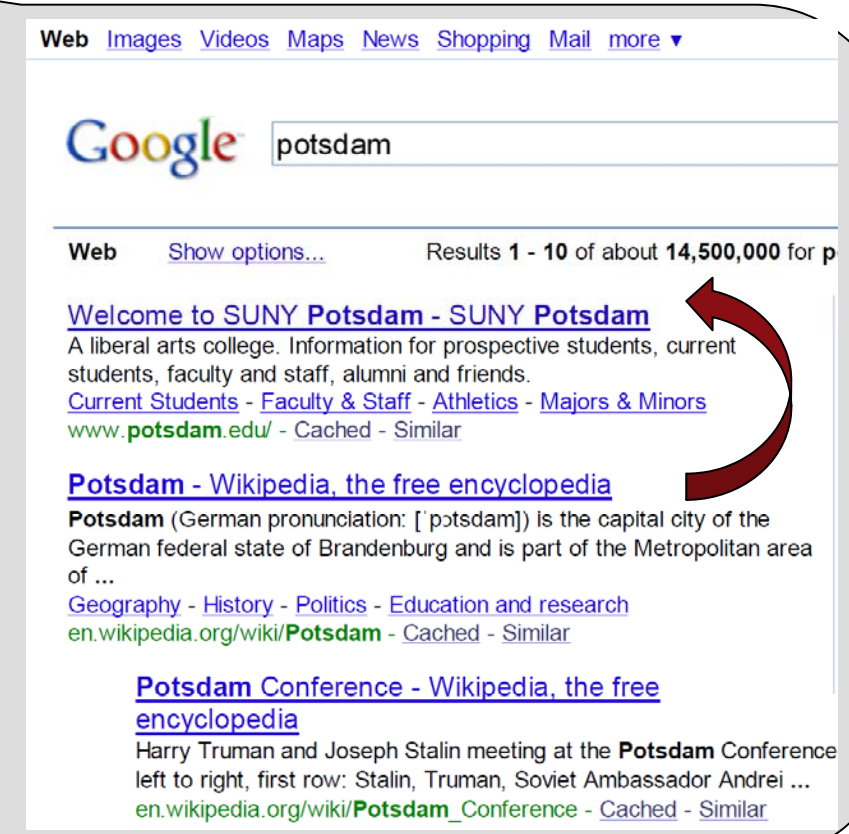
• Ordnung

Andere Lernprobleme

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

• Keine direkten Klassen beobachtet, sondern nur Präferenzen

• z.B Reihenfolge von Suchresultaten aus Clickstreams lernen

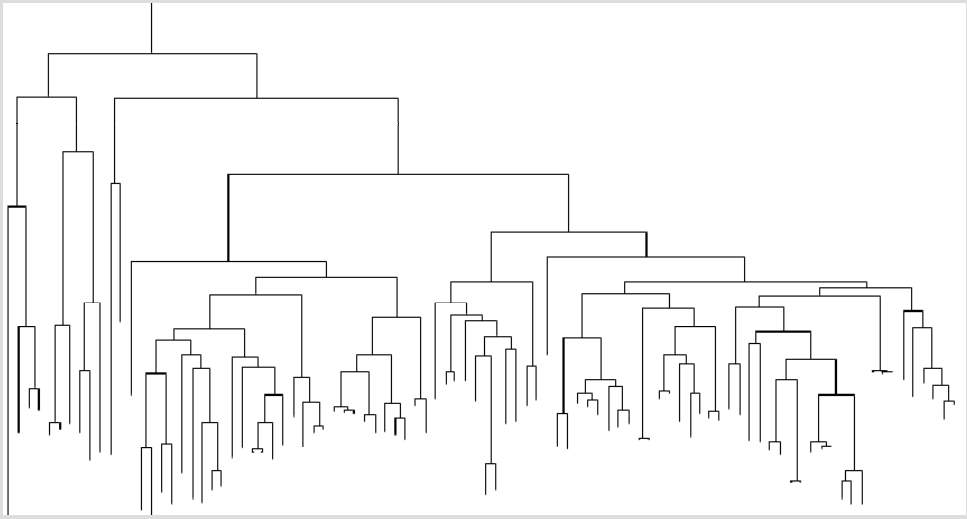


Andere Lernprobleme

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

Hierarchie von Klassen

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



Andere Lernprobleme

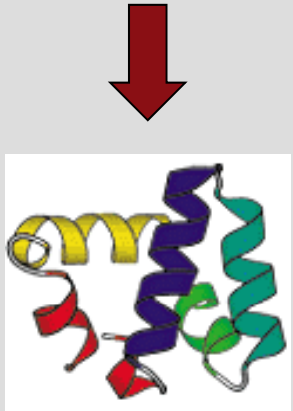
- Ordinale Regression.
- Präferenzlernen.
- Taxonomie-Klassifikation.
- Klassifikation und Regression mit strukturierten Ausgaberäumen.
- Kollaborative Filterung.
- ...

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.
  - $f : X \rightarrow Y$

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

- 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

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

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

Suche nach Hypothese

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

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

Trainingsdaten

Medikamente in der Kombination

Beispiel-Kombinationen

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

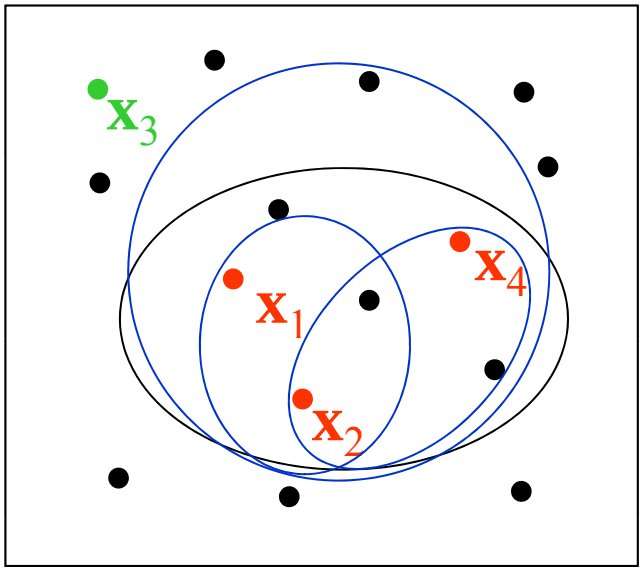
$$\forall i : f(\mathbf{x}_i) = y_i$$

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

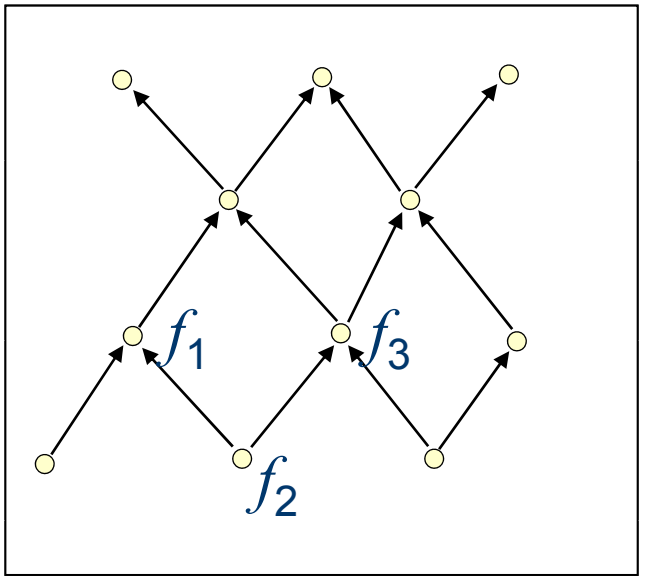
„Genereller-Als“-Ordnung

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

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:

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

- 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: Beoachtungen

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

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

- 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	☹

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

Beispiel: Version Space

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

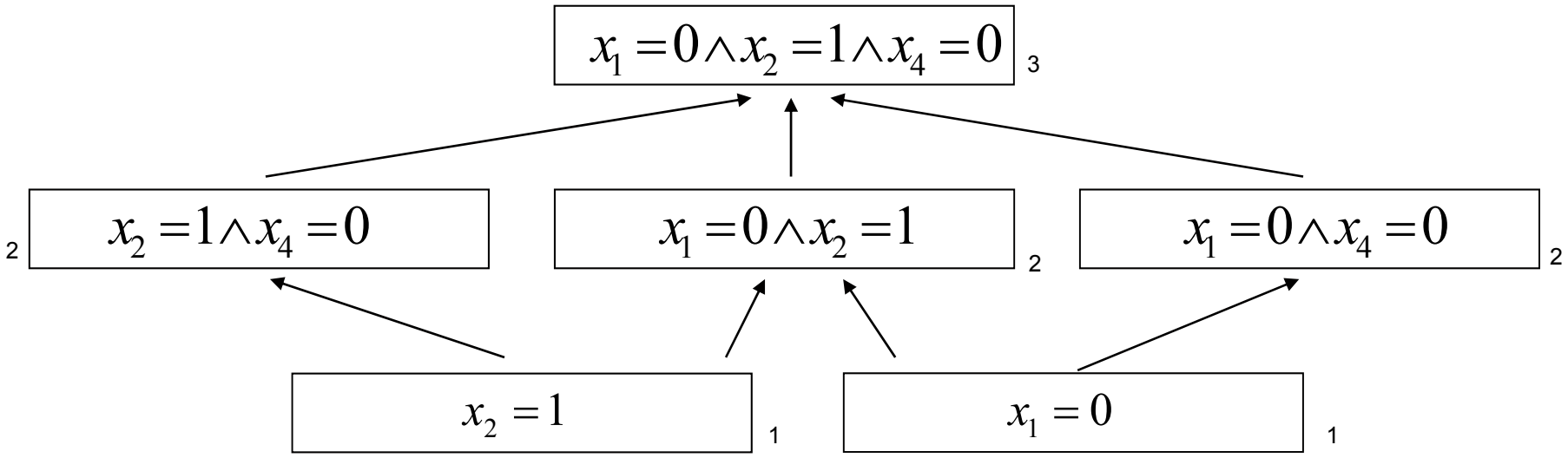
- Welche Hypothesen sind im Version Space?

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

- 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.
  - Jedes Element des Version Space könnte die Daten erzeugt haben / die korrekte Hypothese sein

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

- Verlustfunktionen: Grad der Konsistenz mit Trainingsdaten
- A-Priori-Wahrscheinlichkeit (Prior) über Modelle.
- Wahrscheinlichstes Modell gegeben Daten.

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: Klassifikationsprobleme, 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

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

- 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

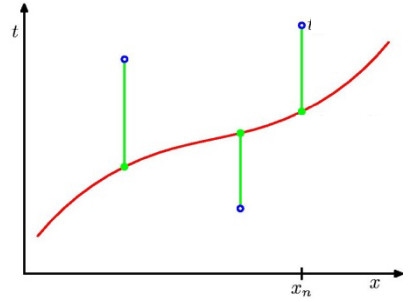
Verlustfunktion

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

- 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
  - $f_{\mathbf{w}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ \sum_j w_j x_j \geq b \\ \text{☺}, & sonst \end{cases}$

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

    w: Modellparameter
  - mit $w_j \in \{-1,0,+1\}$

Regularisierer: Beispiel

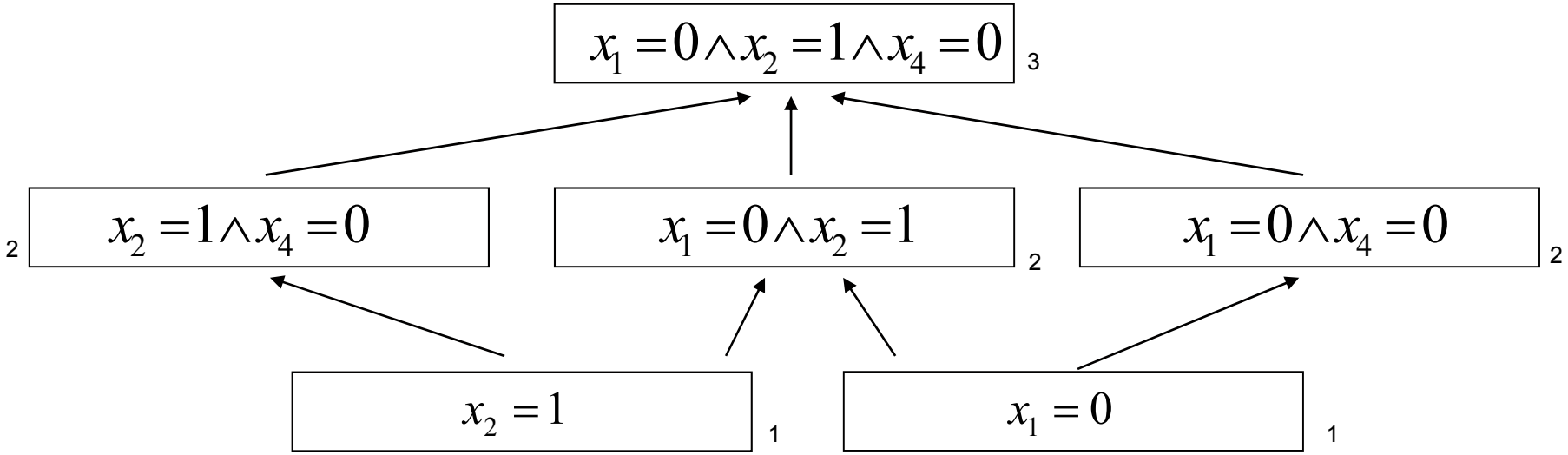
- Linearer Klassifikator
  - $f_{\mathbf{w}}(\mathbf{x}) = \begin{cases} \text{☹}, & wenn\ \mathbf{w}^{\mathrm{T}}\mathbf{x} \geq b \\ \text{☺}, & sonst \end{cases}$
- L_2-Regularisierung:
  - $\lambda \,|\, \mathbf{w} \,|^2 \qquad |\, \mathbf{w} \,|^2 = \sum_i w_i^2$
  - Addiert λ für jedes von null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

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

Optimierungsproblem: Beispiel

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

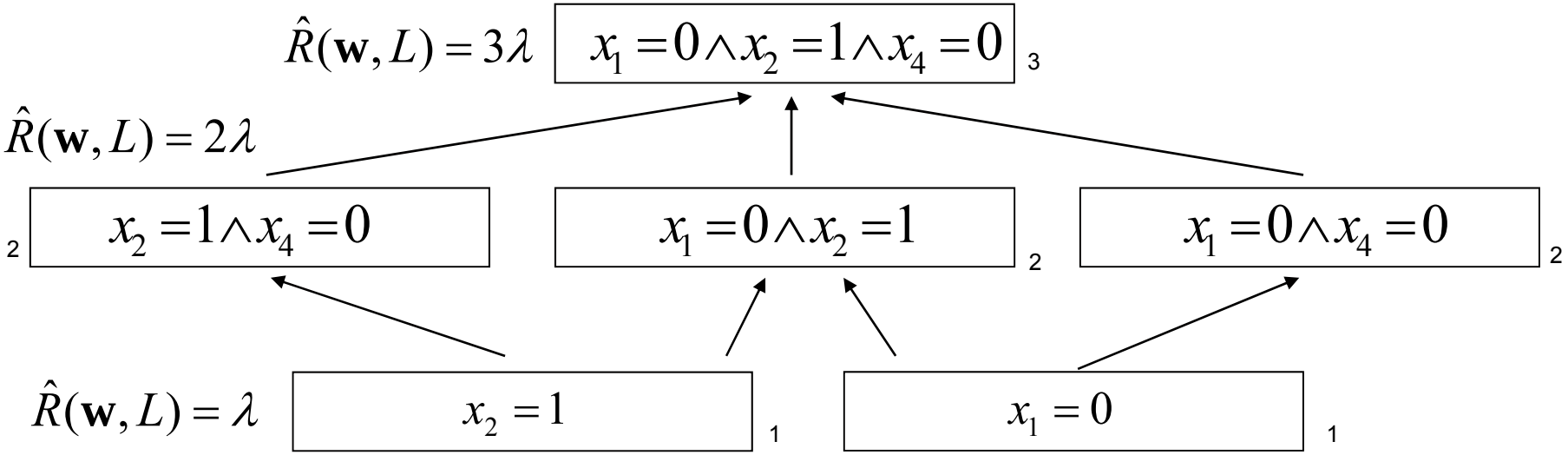
	x_1	x_2	x_3	x_4	x_5	x_6	y
$\mathbf{x}_1$	0	1	0	0	1	1	☹
$\mathbf{x}_2$	0	1	1	0	1	1	☹
$\mathbf{x}_3$	1	0	1	0	1	0	☺
$\mathbf{x}_4$	0	1	1	0	0	0	☹



Optimierungsproblem: Beispiel

- $\hat{R}(\mathbf{w}, L) = \sum_i l(f(\mathbf{x}_i), y_i) + \lambda \,|\, \mathbf{w} \,|^2$
- 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; Lösung existiert manchmal nicht oder 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

Unsicherheit und Wahrscheinlichkeiten

- Viele Verfahren des maschinellen Lernens basieren auf probabilistischen Überlegungen
- Modellvorstellung beim Lernen:
  - Jemand hat echtes f^* nach A-Priori-Wahrscheinlichkeit („Prior“) $p(f)$ gezogen.
  - f^* ist nicht bekannt, aber $p(f)$ reflektiert Vorwissen (was sind wahrscheinliche Modelle?).
  - Trainingseingaben $\mathbf{x}_i$ werden gezogen.
  - Klassenlabels y_i werden nach $p(y_i \mid \mathbf{x}_i, f^*)$ gezogen.
  - Fragestellung Lernen: Gegeben L und $p(f)$, was ist wahrscheinlichstes „echte“ Modell ?
    → Versuche, f^* (ungefähr) zu rekonstruieren

Wahrscheinlichkeitstheorie

- Zufallsexperiment: definierter Prozess, in dem ein Elementarereignis ω erzeugt wird.
- Ereignisraum Ω: Menge aller möglichen Elementar-ereignisse.
- Ereignis A: Teilmenge des Ereignisraums.
- Wahrscheinlichkeitsfunktion p: Funktion, die Wahrscheinlichkeitsmasse auf Ereignisse $A \subseteq \Omega$ verteilt.

Wahrscheinlichkeitstheorie

- Gültige Wahrscheinlichkeitsfunktion p (Kolmogorow-Axiome)
  - Wahrscheinlichkeit von Ereignis $A \subseteq \Omega$: $0 \leq p(A) \leq 1$
  - Sicheres Ereignis: $p(\Omega) = 1$, und $p(\varnothing) = 0$
  - Für die Wahrscheinlichkeit zweier inkompatibler Ereignisse $A \subseteq \Omega, B \subseteq \Omega$ (d.h. $A \cap B = \varnothing$) gilt:

$$p(A \cup B) = p(A) + p(B)$$

Wahrscheinlichkeitstheorie: Beispiel

- Würfeln
  - Ereignisraum $\Omega = \{1,2,3,4,5,6\}$
  - Elementarereignisse haben Wsk $p(\{\omega\}) = 1/6$
  - Ereignis gerade Zahl: $A = \{2,4,6\}$
  - Wahrscheinlichkeit des Ereignisses: $p(A) = 1/2$

Zufallsvariable

- Zufallsvariable X: Abbildung eines elementaren Ereignisses auf einen numerischen Wert

$$X : \Omega \to \mathbb{R}$$
$$X : \omega \mapsto x$$

- Wahrscheinlichkeit, dass ZV einen Wert annimmt

$$p(X = x) = p(\{\omega \in \Omega \mid X(\omega) = x\})$$

- Bei kontinuierlichen ZV oft $p(X = x) = 0$, man betrachtet dann auch Verteilungsfunktion

$$p(X \leq x) = p(\{\omega \in \Omega \mid X(\omega) \leq x\})$$

Zufallsvariable: Beispiel

- Würfeln mit 2 Würfeln
  - Ereignisraum $\Omega = \{(\omega_1, \omega_2) \mid \omega_i \in \{1,2,3,4,5,6\}\}$
  - Elementarereignisse haben Wsk $p(\{(\omega_1, \omega_2)\}) = 1/36$
  - Zufallsvariable: Summe der beide Würfel

$$X((\omega_1, \omega_2)) = \omega_1 + \omega_2$$

  - Wahrscheinlichkeit für Wert der ZV:

$$p(X = 5) = ?$$

Zufallsvariable: Beispiel

- Würfeln mit 2 Würfeln
  - Ereignisraum $\Omega = \{(\omega_1, \omega_2) \mid \omega_i \in \{1,2,3,4,5,6\}\}$
  - Elementarereignisse haben Wsk $p(\{(\omega_1, \omega_2)\}) = 1/36$
  - Zufallsvariable: Summe der beide Würfel

$$X((\omega_1, \omega_2)) = \omega_1 + \omega_2$$

  - Wahrscheinlichkeit für Wert der ZV:

$$p(X = 5) = p(\{(1,4),(2,3),(3,2),(4,1)\})$$

$$= 4/36$$

Gemeinsame und bedingte Wahrscheinlichkeiten

- Oft gibt es mehrere Zufallsvariablen $X, Y, Z, \ldots$
- Gemeinsame Wahrscheinlichkeit:

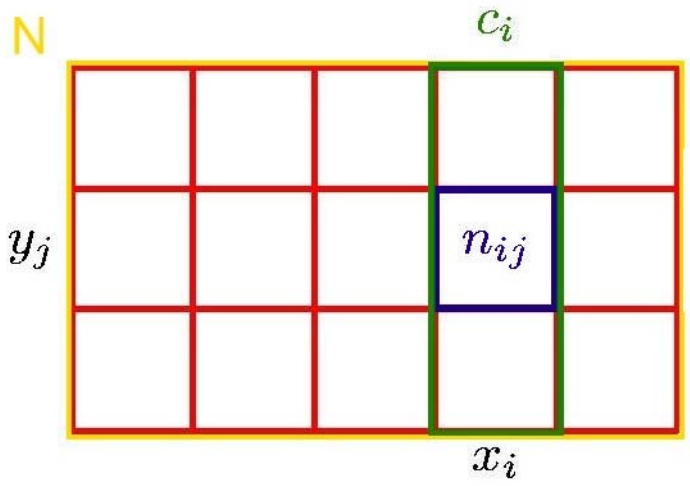
$$p(X = x, Y = y) = p(\{\omega \in \Omega \mid X(\omega) = x \wedge Y(\omega) = y\})$$

- Bedingte Wahrscheinlichkeit

$$p(X = x \mid Y = y) = \frac{p(X = x, Y = y)}{p(Y = y)}$$

Gemeinsame und bedingte Wahrscheinlichkeiten

„Zufälligen Punkt auf Fläche auswählen“



Gemeinsame Wahrscheinlichkeit

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$$

Randwahrscheinlichkeit

$$p(X = x_i) = \frac{c_i}{N}.$$

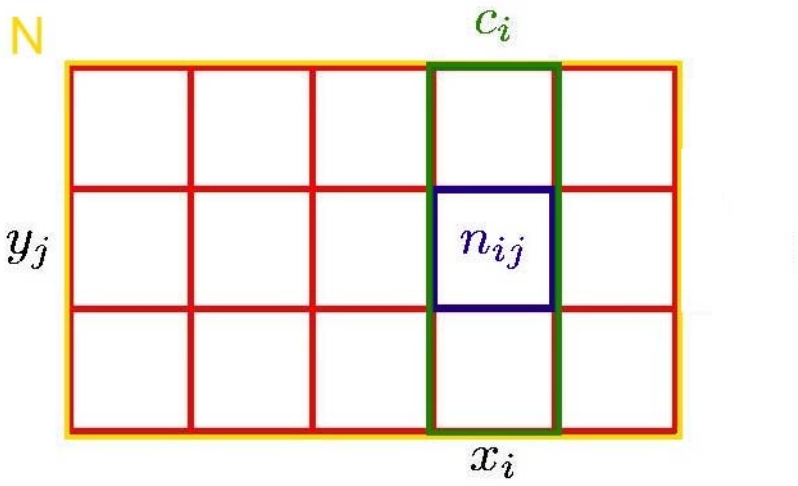
Bedingte Wahrscheinlichkeit

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$$

$$= \frac{p(Y = y_j, X = x_i)}{p(X = x_i)}$$

Wichtige Rechenregeln für Wahrscheinlichkeiten

„Zufälligen Punkt auf Fläche auswählen“



Summenregel

$$p(X = x_i) = \sum_j p(X = x_i, Y = y_j)$$

$$p(X = x_i) = \frac{c_i}{N} = \frac{1}{N} \sum_j n_{ij}$$

$$= \sum_j p(X = x_i, Y = y_j)$$

Produktregel

$$p(Y = y_j, X = x_i) = p(Y = y_j \mid X = x_i)\, p(X = x_i)$$

$$p(Y = y_j, X = x_i) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \frac{c_i}{N} = p(Y = y_j \mid X = x_i)\, p(X = x_i)$$

Wichtige Rechenregeln für Wahrscheinlichkeiten

- Summenregel $$p(X) = \sum_{Y} p(X, Y)$$
- Produktregel $$p(X, Y) = p(Y|X)p(X)$$

Unabhängigkeit von Zufallsvariablen

- Variablen X, Y unabhängig gdw

$$p(X = x_i, Y = y_j) = p(X = x_i)\, p(Y = y_j)$$

- Äquivalente Definition Unabhängigkeit:

$$p(X = x_i | Y = y_j) = p(X = x_i) \quad \text{und} \quad p(Y = y_j | X = x_i) = p(Y = y_j)$$

- Bedingte Unabhängigkeit

$$p(X = x_i, Y = y_i \mid Z = z_k) = p(X = x_i \mid Z = z_k)\, p(Y = y_i \mid Z = z_k)$$

Bayessche Regel

- Bayessche Regel:

$$p(X \mid Y) = \frac{p(Y \mid X)\,p(X)}{p(Y)}$$

- Beweis einfach:

$$p(X \mid Y) = \frac{p(X,Y)}{p(Y)} = \frac{p(Y \mid X)\,p(X)}{p(Y)}$$

Definition bedingte Wahrscheinlichkeit

Produktregel

- Wichtige Grundeinsicht für das maschinelle Lernen: Erlaubt den Rückschluss auf *Modellwahrscheinlichkeiten* gegeben *Wahrscheinlichkeiten von Beobachtungen*

Bayessche Regel

- Modellwahrscheinlichkeit gegeben Daten und Vorwissen

$$p(Modell \mid Daten) = \frac{p(Daten \mid Modell)\, p(Modell)}{p(Daten)}$$

$$\propto p(Daten \mid Modell)\, p(Modell)$$

$p(Daten)$ konstant, unabhängig von *Modell*

Likelihood: wie gut erklärt Modell die Daten?

Prior: wie wahrscheinlich ist Modell a priori?

Maximum-A-Posteriori-Hypothese

- Wahrscheinlichstes Modell gegeben die Daten.
  - $f_{MAP} = \arg\max_{f_{\mathbf{w}}} p(f_{\mathbf{w}}|L)$

$$= \arg\max_{f_{\mathbf{w}}} \frac{p(L|f_{\mathbf{w}})p(f_{\mathbf{w}})}{p(L)}$$

Anwendung Bayes'sche Regel

$$= \arg\max_{f_{\mathbf{w}}} p(L|f_{\mathbf{w}})p(f_{\mathbf{w}})$$

$$= \arg\min_{f_{\mathbf{w}}} -\log P(L|f_{\mathbf{w}}) - \log p(f_{\mathbf{w}})$$

Log-Likelihood

Log-Prior

$\Rightarrow$ Optimierungskriterium bestehend aus log-likelihood und log-prior

$\mathbf{w}$ Parameter des Modells $f_{\mathbf{w}}(\mathbf{x})$

Log-Likelihood

- Wie wahrscheinlich sind die Daten gegeben das Modell?
  - $-\log p(L|f_{\mathbf{w}}) = -\log p(y_1,...,y_N|f_{\mathbf{w}},\mathbf{x}_1,...,\mathbf{x}_N)$
- Annahme: Datenpunkte sind unabhängig gezogen.
  - $$-\log p(y_1,...,y_N|f_{\mathbf{w}},\mathbf{x}_1,...,\mathbf{x}_N)$$
    $$= -\log \prod_i p(y_i|f_{\mathbf{w}},\mathbf{x}_i)$$
    $$= \sum_i -\log p(y_i|f_{\mathbf{w}},\mathbf{x}_i)$$

Annahme: spezielle Exponential-verteilung

Log-Likelihood

- Wie wahrscheinlich sind die Daten gegeben das Modell?
  - $-\log p(L|f_{\mathbf{w}}) = -\log p(y_1,...,y_N|f_{\mathbf{w}},\mathbf{x}_1,...,\mathbf{x}_N)$
- Annahme: Datenpunkte sind unabhängig gezogen.
  - $$-\log p(y_1,...,y_N|f_{\mathbf{w}},\mathbf{x}_1,...,\mathbf{x}_N)$$
    $$= -\log \prod_i p(y_i|f_{\mathbf{w}},\mathbf{x}_i)$$
    $$= \sum_i -\log p(y_i|f_{\mathbf{w}},\mathbf{x}_i)$$

Annahme: spezielle Exponential-verteilung

$$p(y_i|f_{\mathbf{w}},\mathbf{x}_i) = \frac{1}{Z} e^{-l(f(\mathbf{x}_i),y_i)}$$

Verlustfunktion

Normalisierer

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

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

$$c, \overline{c} \neq 0$$

Log-Likelihood

- Wie wahrscheinlich sind die Daten gegeben das Modell?
  - $-\log p(L|f_{\mathbf{w}}) = -\log p(y_1,...,y_N|f_w,\mathbf{x}_1,...,\mathbf{x}_N)$
- Annahme: Datenpunkte sind unabhängig gezogen.

$$
\begin{aligned}
&-\log p(y_1,...,y_N|f_{\mathbf{w}},\mathbf{x}_1,...,\mathbf{x}_N) \\
&= -\log \prod_i p(y_i|f_{\mathbf{w}},\mathbf{x}_i) \\
&= \sum_i -\log p(y_i|f_{\mathbf{w}},\mathbf{x}_i) \\
&= \sum_i -\log \frac{e^{-l(f(\mathbf{x}_i),y_i)}}{Z} \\
&= \sum_i l(f_{\mathbf{w}}(\mathbf{x}_i),y_i) + const
\end{aligned}
$$

Konstanter Faktor (unabhängig von f)

- Negative Log-Likelihood entspricht Verlustterm!

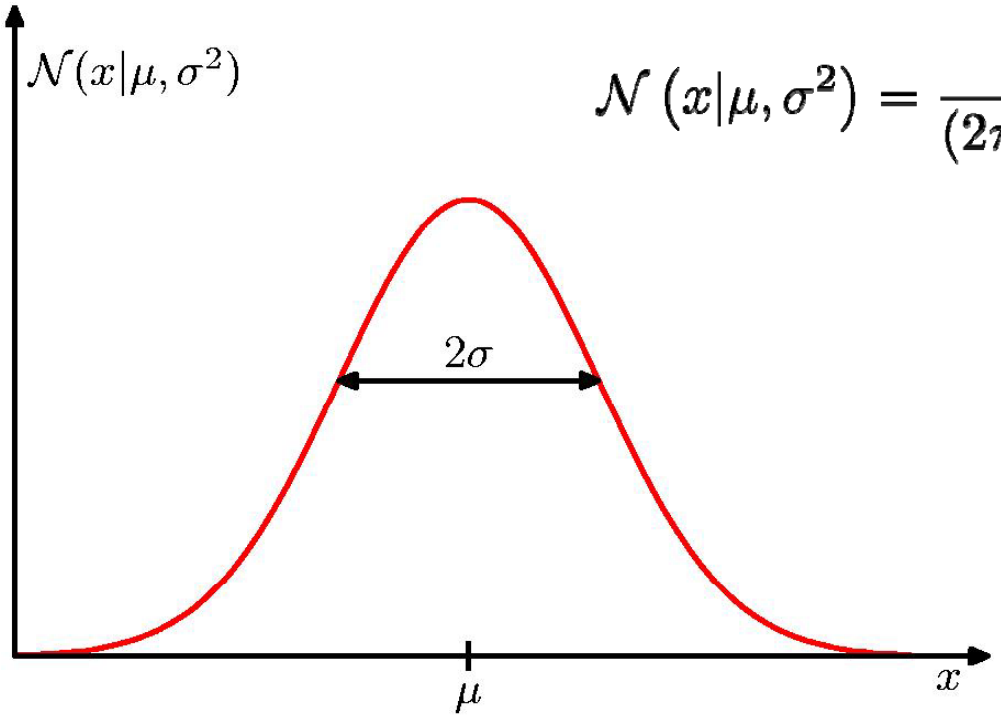
Prior: Gaußverteilung, Normalverteilung

Verteilung für reelle (oder vektor-wertige) Zufallsvariable x

Streut um Mittelwert μ mit Varianz σ^2

Dichtefunktion

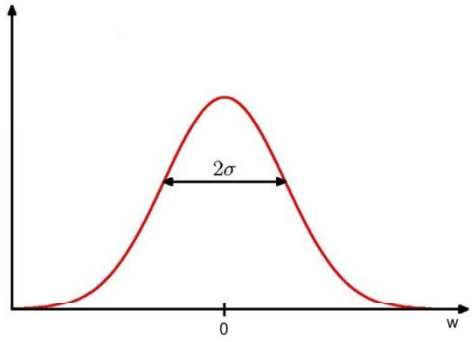
$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$



A-Priori-Wahrscheinlichkeit (Prior)

- Annahme: Modellparameter normalverteilt mit Mittelwert $\mu = 0$

  - $$p(f_{\mathbf{w}}) = \mathcal{N}(\mathbf{w} \mid 0, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}|\mathbf{w}|^2}$$



- Negativer Log-Prior:

  - $$-\log p(f_{\mathbf{w}}) = \frac{1}{2\sigma^2} |\mathbf{w}|^2 + const$$

Konstanter Faktor (unabhängig von f)

- Negativer Log-Prior = Regularisierer!

A-Posteriori-Wahrscheinlichkeit (Posterior)

- Wahrscheinlichstes Modell gegeben Vorwissen und Daten.
  - $$f_{MAP} = \arg\max_{f_{\mathbf{w}}} p(f_{\mathbf{w}}|L)$$
    $$= \arg\min_{f_{\mathbf{w}}} -\log p(L|f_{\mathbf{w}}) - \log p(f_{\mathbf{w}})$$
    $$= \arg\min_{f_{\mathbf{w}}} \sum_i l(f_{\mathbf{w}}(\mathbf{x}_i), y_i) + \lambda |\mathbf{w}|^2$$

    (Folie 60 zusammen mit 63 & 65, $\lambda = \frac{1}{2\sigma^2}$)

- ArgMin über regularisierte Verlustfunktion!
- Rechtfertigung für Optimierungskriterium?
  - **Wahrscheinlichste Hypothese (*MAP-Hypothese*).**

Zusammenfassung

- Klassifikation, Regression, Modelle für Vorhersage.
- Parametrisierung von Modellen (Parametervektor **w**)
- Hypothesenraum: Raum aller Modelle.
- Version Space: Modelle, die mit Trainingsdaten konsistent sind.
- Lernen (Parameterschätzung) aus Trainingsdaten.
  - Optimierungskriterium: Verlust über Trainingsbeispiele plus Regularisierungsterm.
  - Minimum des Kriteriums: wahrscheinlichstes Modell gegeben Daten und Prior.

Übungen

- Erstes Übungsblatt: Ausgabe morgen, Besprechung kommende Woche
- Sie können für einzelne Aufgaben votieren
- Sie müssen für 2/3 aller Aufgaben votieren