



Teil 5



Maschinelles Lernen

Definitionen und Abgrenzungen

➤ Was ist Lernen?

- ⇒ Zentrale Fähigkeit von „intelligenten Systemen“ in Natur und KI
- ⇒ Zielgerichtete Veränderung von Wissen und/oder Verhaltensweisen durch Erfahrung
- ⇒ Verallgemeinerbarkeit auf neue Situationen
- ⇒ Viele Formen des Lernens:
 - ✧ motorisches Lernen, Regellernen, Sprachlernen, Lernen mit Lehrern, Lernen in der Entwicklung, ...

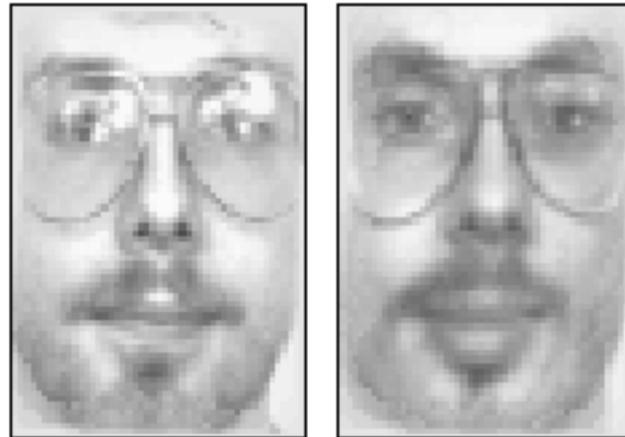
➤ Was ist kein Lernen?

- ⇒ festes Programmieren von Lösungen (keine Erfahrung)
- ⇒ Einfaches Speichern von Daten (keine Verallgemeinerung)

Beispiel: Gesichtserkennung

➤ Erkennung anhand von Gesichtern:

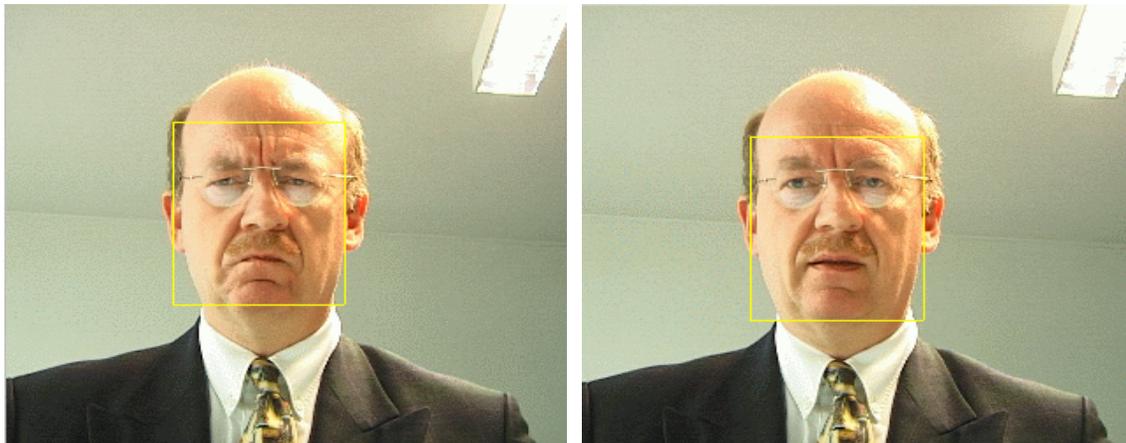
- ➔ Person
- ➔ Geschlecht
- ➔ Stimmung



Bsp.: Wiskott et al.

m	m	m	m	g	g	g	g
m	m	m	m	.	g	g	g
m	m	m	m	g	g	g	g
m	m	m	m	g	g	g	g
m	m	m	m	b	b	b	b
m	m	m	m	b	b	b	b
m	m	m	m	.	b	b	.

attributes determined: person is male, has glasses, and is bearded



Bsp.: SmartKom System

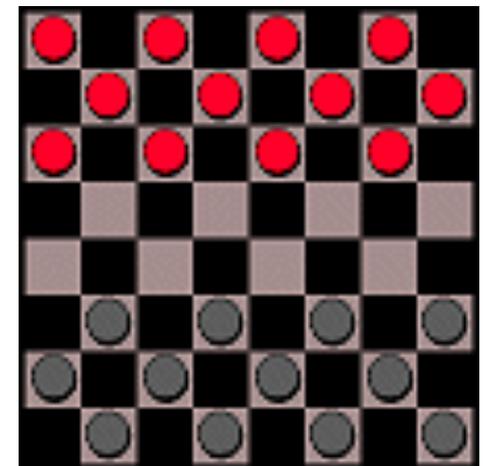
Beispiel: automatische Fahrzeugnavigation

- Ziel selbständige Fahrzeuge in z.T. unbekanntenen Umgebungen
- Viele zu lernende Teilprobleme
 - ⇒ Steuerung und Aktorik
 - ⇒ Situationserkennung
 - ⇒ Planung und Navigation
 - ⇒ ...



Beispiel: Spiele

- Backgammon, Schach, Mühle etc.
- Gutes „Spielfeld“ für neue Verfahren
- Typische Domäne von Menschen
- Beispiel: TD-Gammon nutzt temporal difference (TD) Algorithmen und trainiert, während es mit sich selbst spielt
- Auf dem Treffen der American Association of Artificial Intelligence wurde 1998 ein Großmeister (der damalige Weltmeister) in 99 von 100 Spielen von einer Maschine besiegt.



Beispiel: Handschriftenerkennung



- Erkennen von handgeschriebenen Zeichen hat bis 1995 sehr schlecht funktioniert.

- Künstliche Neuronale Netze haben dies im Apple Newton wesentlich verbessert.
 “...vastly improved hand-writing recognition...” (BYTE May 1996)

- 2003: Die aktuelle Version von Mac OS besitzt eine integrierte Unterstützung für Stifteingabe, die auf der Schrifterkennung des Newton basiert.

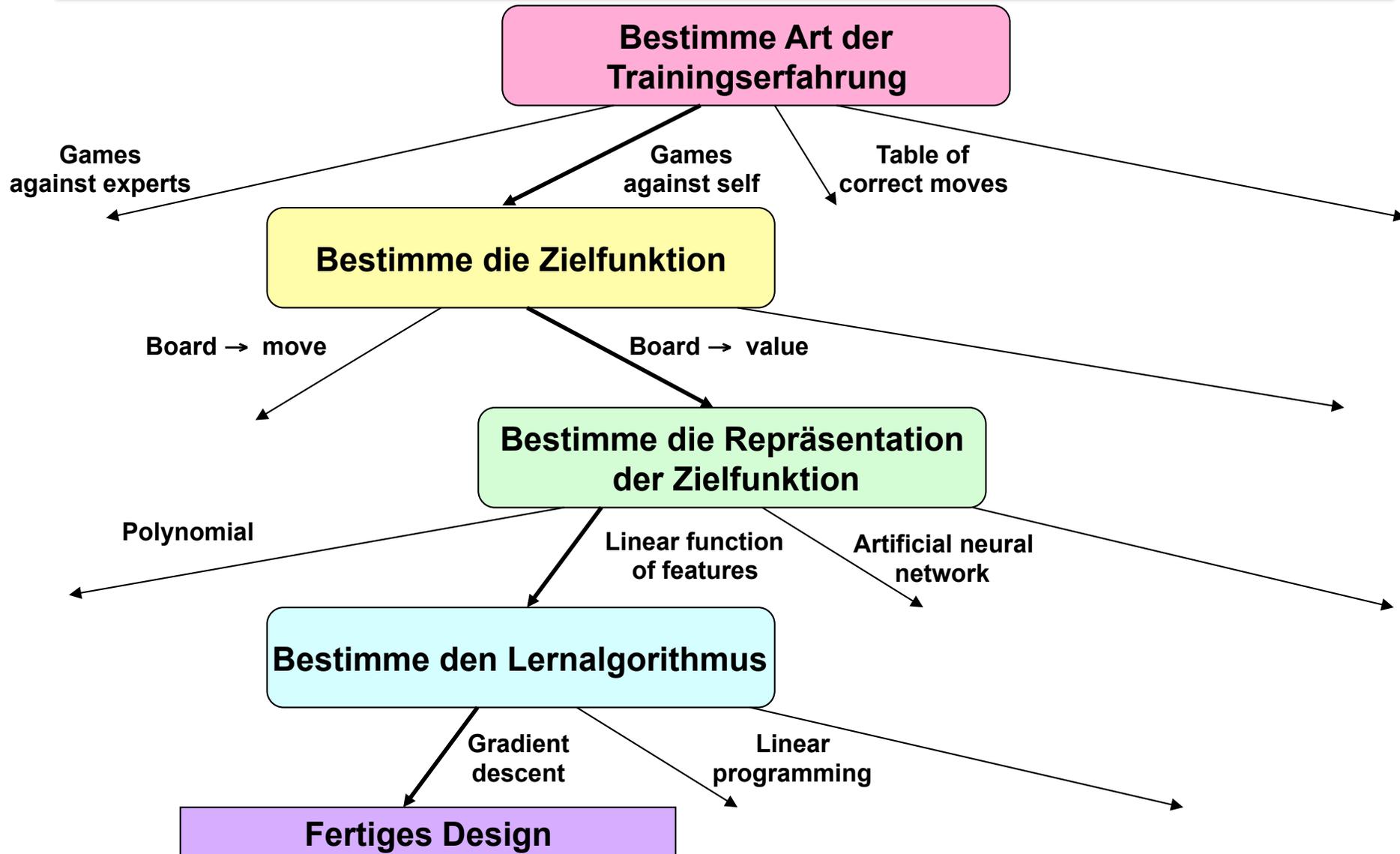
Wozu maschinelles Lernen?

- **Anwendungsfelder:**
 - ⇒ Data Mining:
Extraktion von Wissen aus Daten
 - ⇒ Selbst-adaptierende Programme/Filter:
dynamische Anpassung an Nutzer und Situationen
 - ⇒ Aktionslernen:
Robotik, Steuerungen, Unterstützung bei Entscheidungen
 - ⇒ Anwendungen, die schwer programmierbar sind (explizites Modell nicht verfügbar oder zu teuer):
Sprachverarbeitung, Fahrzeugsteuerung
- **Besseres Verständnis des menschlichen Lernens und Lehrens**
 - ⇒ Kognitionswissenschaften:
Theorien des Wissenserwerbs (z.B. durch Übung)
 - ⇒ Technische Umsetzung:
in Lernsysteme, Recommender Systeme etc.

Spezifikation von Lernproblemen

- Lernen = Verbesserung der Leistung eines Systems
 - ⇒ bei einer Aufgabe A,
 - ⇒ in Bezug auf ein Leistungsmaß/eine Zielfunktion Z,
 - ⇒ basierend auf der vorhergehenden Erfahrung E.
- Beispiel: Schach spielen
 - ⇒ A: spiele Schach und gewinne
 - ⇒ Z: Prozentsatz der gewonnenen Spiele in der ersten Liga
 - ⇒ E: Menge bekannter Spiele
- Zu lösende Fragen (Modellierungsentscheidungen):
 - ⇒ Wie genau soll die Erfahrung gesammelt werden?
 - ⇒ Was genau soll eigentlich gelernt werden?
 - ⇒ Wie wird das Gelernte repräsentiert?
 - ⇒ Welcher Lernalgorithmus soll angewandt werden?

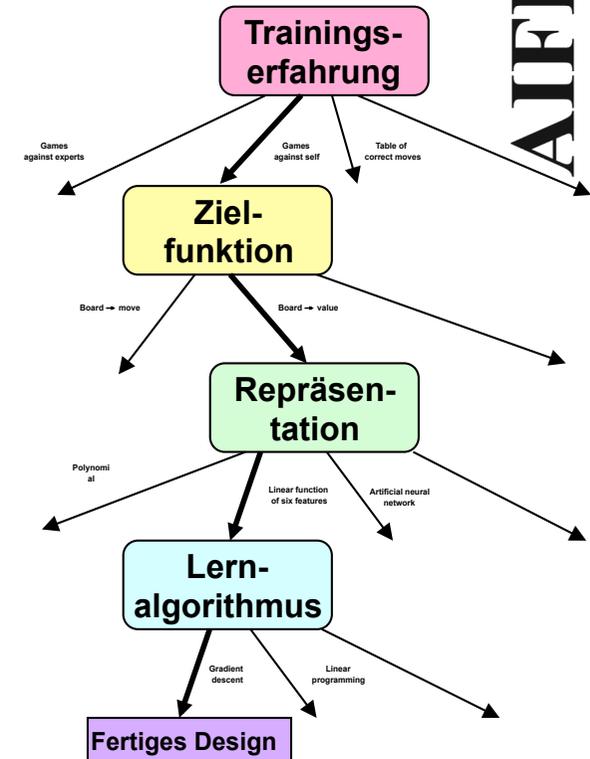
Problemlösung mit maschinellem Lernen



Problemlösung mit maschinellem Lernen

➤ Bemerkungen:

- ⇒ Dieses Vorgehen ist sehr allgemein.
- ⇒ Oft wird unter maschinellem Lernen nur der Teil „Lernalgorithmus“ verstanden, es geht aber um die ganze Pipeline.
- ⇒ Nur wenn alle Teile zusammenpassen ist ein gutes Resultat zu erwarten.
- ⇒ Dieses Modell ist modular und Verfahren auf den verschiedenen Ebenen können miteinander kombiniert werden.
- ⇒ Je nach zu lösendem Problem, kann es sein, dass es in mehrere Lernprobleme zerlegt werden muss, die durch unterschiedliche Ansätze gelöst werden.



Trainingserfahrung

➤ Designentscheidungen:

⇒ Erzeugung von Beispielen

- ✧ Beispiele aus Datenbanken/-sammlung
- ✧ Erzeugung durch das System
- ✧ Vorgabe durch Experten

⇒ Welche Vorgabe von außen?

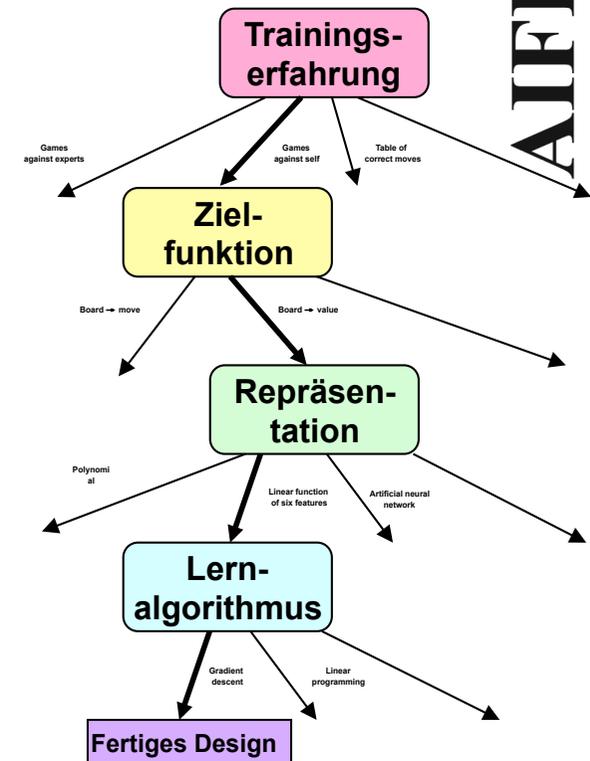
- ✧ Überwachtes Lernen (Bewertung durch Lehrer)
- ✧ Unüberwachtes Lernen (keine Vorgabe)
- ✧ Reinforcement Learning (Bewertung über Erfolg/Misserfolg einer Serie von Entscheidungen)

⇒ Wann gibt es Vorgaben?

- ✧ Getrennte Lern- und Kannphase
- ✧ Kontinuierliches Lernen
- ✧ Lernen mit Vergessen

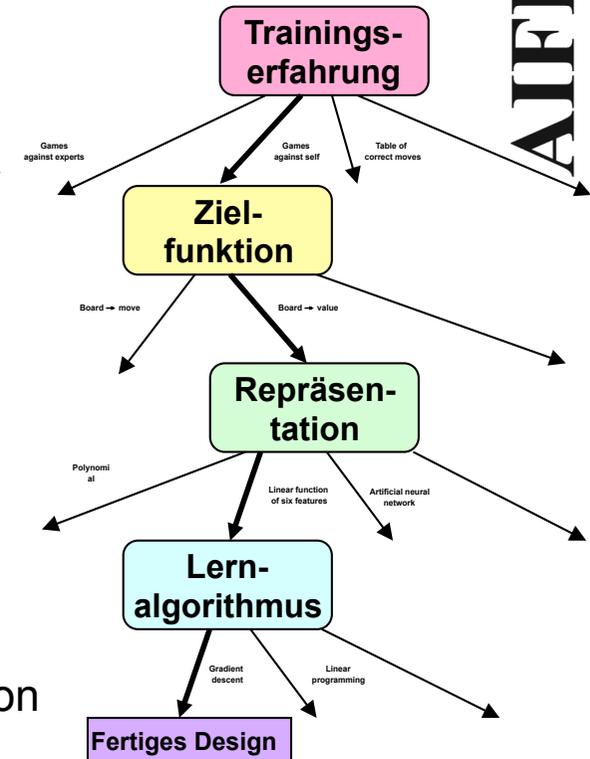
⇒ Wann wird gelernt?

- ✧ In jedem Schritt (Learning by Pattern)
- ✧ Nach einigen Schritten (Learning by Block)
- ✧ Nach einem kompletten Satz von Eingaben (Learning by Epoch)



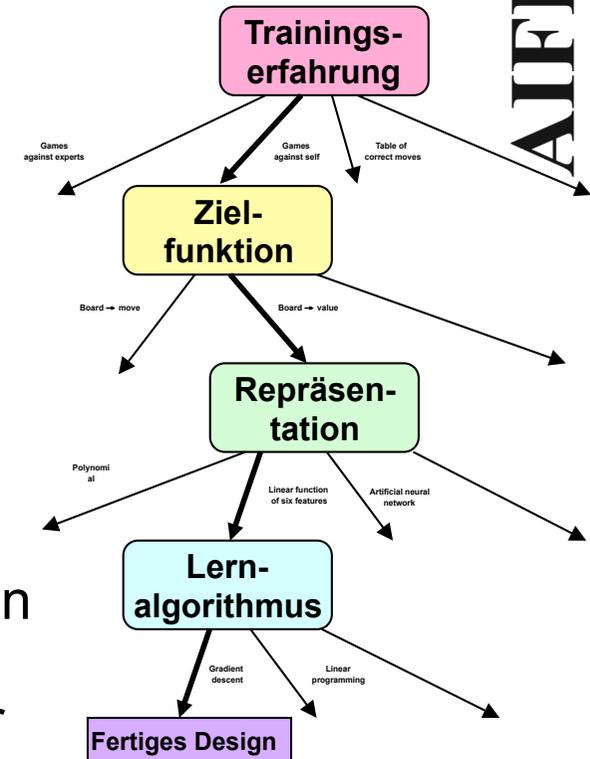
Zielfunktion

- oft „schwarze Kunst“, welche Kodierung für konkreten Fall am besten geeignet ist
- meist wird hier das Problem als Problem der Mustererkennung aufgefasst, also
 - ⇒ Klassifikation von Mustern
 - ✧ Ist ein Produkt in Ordnung oder nicht?
 - ✧ Ist ein Patient krank oder gesund?
 - ✧ Ist eine handgeschriebene Ziffer eine 1,2,3 ... ?
 - ⇒ Bewertung von Mustern
 - ✧ Oft: Generalisierung (Schätzen von Werten für bestimmte Beobachtungen bei vorgegebenen Mustern. Rekonstruktion einer Funktion)
 - ✧ Was ist ein vernünftiger Preis für dieses Haus / Auto?
 - ✧ Wie gesund/krank ist ein Patient?
 - ✧ In der klassischen Statistik sind das Regression, Inter- sowie Extrapolation. Für hochdimensionale Probleme tritt in der Praxis meist Extrapolation auf
- Die Güte der Zielfunktion ist nicht unabhängig von der Wahl des Lernalgorithmus



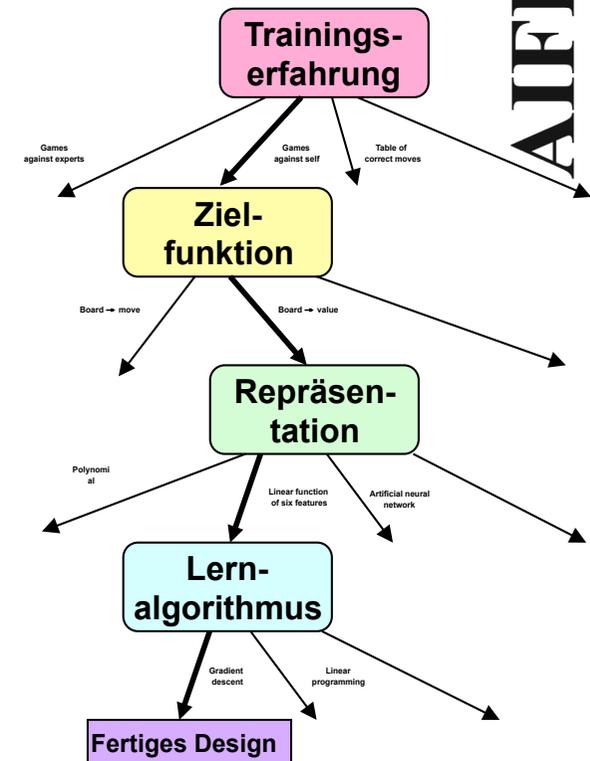
Repräsentation

- Zwei Lager: symbolisch und subsymbolisch
- Symbolisch
 - ⇒ Beispiele: Regeln, Entscheidungsbäume, logische Formeln, Beispielfälle
 - ⇒ Vorteil: Erklärungsmöglichkeit
- Subsymbolisch
 - ⇒ Beispiele: Statistik, Polynome, neuronale Netze, „Genstrings“
 - ⇒ Vorteil: Leistungsfähigkeit
- Repräsentation schränkt Auswahl der Lernalgorithmen stark ein
- Neben der Repräsentation ist auch die Kodierung der Eingangsdaten entscheidend
 - ⇒ Beispiel: Kodierung des Alters von Patienten als Integer, als reelle Werte, als binäre Klassenvariablen („Alter 0-10“, „Alter 11-30“, „über 30“)
 - ⇒ Ungeeignete Kodierungen können aus einfachen Lernproblemen schwierige machen!



Lernalgorithmus

- Eigentlicher Kern des Maschinellen Lernens
- Viele Verfahren aus verschiedenen Bereichen (Statistik, Logik, Neurowissenschaften, Optimierung, ...)
 - ⇒ Schätzverfahren
 - ⇒ Induktives Schließen
 - ⇒ Case-Based Reasoning
 - ⇒ Backpropagation
 - ⇒ LMS-Verfahren
 - ⇒ Genetische und evolutionäre Algorithmen
 - ⇒ Support-Vektor-Maschinen
 - ⇒ Selbstorganisierende Karten

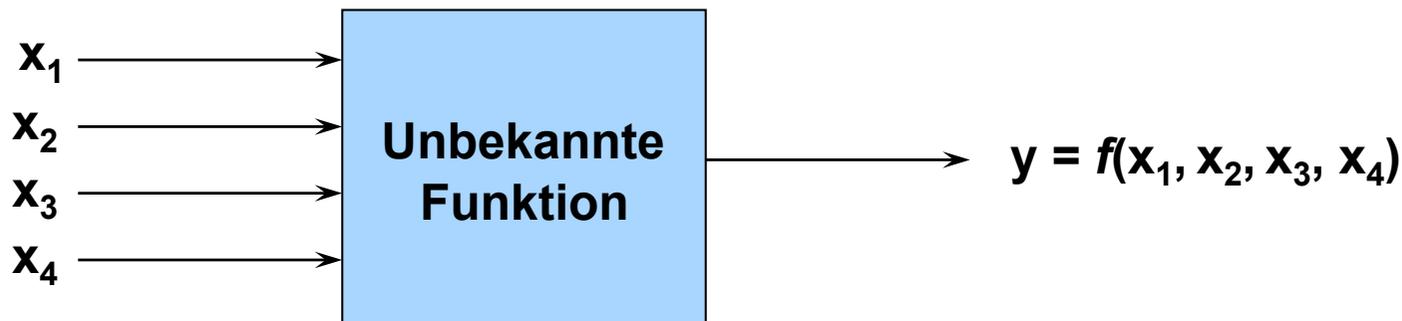


Herausforderungen beim Maschinellen Lernen

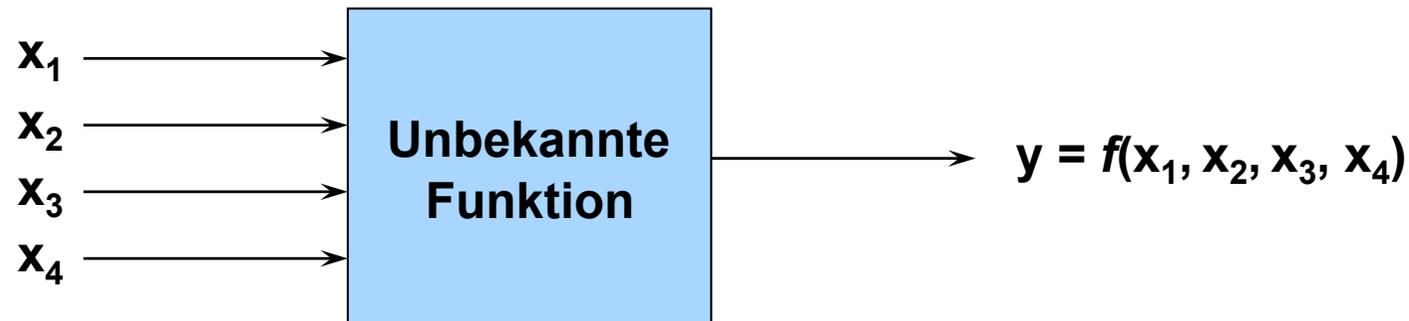
- Welche Verfahren eignen sich wann, um Funktionen anzunähern?
- Entscheidungen beim Design einer Lösung...
 - ⇒ Anzahl der Beispiele?
 - ⇒ Komplexität des Problems
- Wie wird die Lösung des Problems durch Umwelt beeinflusst?
 - ⇒ Verrauschte Daten
 - ⇒ Mehrere Datenquellen
- Was sind die theoretischen Grenzen der Lernbarkeit?
- Wie kann Vorwissen genutzt werden?
- Welche Anregungen können wir aus der Natur abschauen?
- Wie können Systeme selbstständig Repräsentationen finden?

Lernen aus Beispielen

- Gegeben: Trainingsbeispiele $\langle x, f(x) \rangle$ einer unbekanntes Funktion $f(x)$
- Gesucht: Gute Approximation von f
- Einordnung: überwachtes Lernen/Mustererkennung
- Anwendungen:
 - ⇒ Diagnose von Krankheiten
 - ✧ x = Patientendaten (Geschlecht, Alter, Erkrankung, ...)
 - ✧ f = Krankheit (oder vorgeschlagene Therapie)
 - ⇒ Risikobewertung
 - ✧ x = Eigenschaften von Kunden (Demographie, Unfälle, Vorgeschichte, ...)
 - ✧ f = Risikoeinschätzung
 - ⇒ Part-of-speech Tagging
 - ⇒ ...



Ein Lernproblem



Example	x_1	x_2	x_3	x_4	y
0	0	1	1	0	0
1	0	0	0	0	0
2	0	0	1	1	1
3	1	0	0	1	1
4	0	1	1	0	0
5	1	1	0	0	0
6	0	1	0	1	0

- Wir schreiben $A \rightarrow B$ für die Menge aller Funktionen, die Elemente von A auf Elemente von B abbilden
- $x_i \in X_i, y \in Y, f \in (X_1 \times X_2 \times X_3 \times X_4) \rightarrow Y$
- Definitionsbereich von $x = (x_1, x_2, x_3, x_4)$ ist $X = (X_1 \times X_2 \times X_3 \times X_4)$, d.h. $f \in X \rightarrow Y$
- Beispiel: $X_i = Y = \{0, 1\}$

Hypothesen-Raum (unbeschränkter Fall)

- Wieviel mögliche Funktionen f (Hypothesen) gibt es?
- $|A \rightarrow B| = |B|^{|A|}$
- $|\{0,1\} \times \{0,1\} \times \{0,1\} \times \{0,1\} \rightarrow \{0,1\}| = 2^{16} = 65536$ mögliche f
- Naiver Ansatz: Streiche mit jedem Beispiel alle nicht passenden f
 - ⇒ Dazu müssen alle möglichen Eingaben betrachtet werden
 - ⇒ Nach 7 Beispielen bleiben noch $2^9 = 512$ Hypothesen für f (von 65536)
 - ⇒ Keine Vorhersage für unbekannte Beispiele
 - ⇒ kein echtes (verallgemeinerndes) Lernen
 - ⇒ eher: Look-Up-Table

Example	x_1	x_2	x_3	x_4	y
0	0	0	0	0	?
1	0	0	0	1	?
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	?
8	1	0	0	0	?
9	1	0	0	1	1
10	1	0	1	0	?
11	1	0	1	1	?
12	1	1	0	0	0
13	1	1	0	1	?
14	1	1	1	0	?
15	1	1	1	1	?

Spezialfall für Lernen aus Beispielen: Begriffslernen

- Ein **Begriff** (engl.: concept) bezeichnet eine Menge von Entitäten (Begriffsumfang) mit gemeinsamen Eigenschaften (Begriffsinhalt).
- Beim **Begriffslernen** (concept learning) enthält der Wertebereich der zu lernenden Funktion f nur die Werte 1 (wahr, gehört zum Begriff) und 0 (falsch, gehört nicht zum Begriff).

Lernen eines Begriffs, Beispiel EnjoySport

- Begriff EnjoySport umfasst die Tage, an denen Harry gern Sport treibt
- Beispiele
 - ⇒ Ähnlich zur Definition von Datentypen, oft Aufzählungs-Datentypen
 - ⇒ Hier: \in 6 Attribute:

Sky \in {Rainy, Sunny} Temp \in {Warm, Cold}
 Humidity \in {Normal, High} Wind \in {None, Mild, Strong}
 Water \in {Cool, Warm} Forecast \in {Same, Change}

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
0	Sunny	Warm	Normal	Strong	Warm	Same	Yes
1	Sunny	Warm	High	Strong	Warm	Same	Yes
2	Rainy	Cold	High	Strong	Warm	Change	No
3	Sunny	Warm	High	Strong	Cool	Change	Yes

- Ziel: Finde Beschreibung des Begriffes, generalisiere für unbekannte Daten

Repräsentierung der Hypothesen

- Viele Möglichkeiten
- Vorschlag: Jede Hypothese ist eine Konjunktion von Attributbeschreibungen:
 - ⇒ Für jedes Attribut werden Constraints/Bedingungen angegeben:
 - ✧ Spezifische Werte: z.B. Water = Warm
 - ✧ oder: Wert ist egal : z.B. Water = ?
 - ✧ oder: gar kein Wert erlaubt : z.B. Water = \emptyset

- Beispiel-Hypothese für *EnjoySport*

Sky	AirTemp	Humidity	Wind	Water	Forecast
<Sunny	?	?	Strong	?	Same>

- ⇒ Ist diese Hypothese konsistent mit den Trainingsbeispielen?
 - ⇒ Wie sehen Hypothesen aus, die konsistent mit den Beispielen sind?
- Hypothese $h \in H$
- H ist die Menge aller möglichen Hypothesen $H \subseteq X \rightarrow \{0,1\}$
- Bemerkung: $\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle = \langle ?, ?, ?, ?, \emptyset \rangle, \dots$
 sobald ein \emptyset vorkommt entspricht die Hypothese der Funktion $f(x)=0$
- Anzahl der möglichen Hypothesen: $(3 \cdot 3 \cdot 3 \cdot 4 \cdot 3) + 1 = 973 = |H|$
- Bemerkung: $|H|=973 \ll |X \rightarrow \{0,1\}|=2^{(2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 \cdot 2)} = 2^{96} \approx 7,9 \cdot 10^{27}$

Lernen des Konzepts *EnjoySports*

➤ Gegeben:

Menge D von Trainingsbeispielen $\langle x, f(x) \rangle$,

⇒ die gegebene Tage mit den Attributen Sky, AirTemp, Humidity, Wind, Water, Forecast beschreiben (x)

⇒ $X = \{\{\text{Rainy, Sunny}\} \times \{\text{Warm, Cold}\} \times \{\text{Normal, High}\} \times \{\text{None, Mild, Strong}\} \times \{\text{Cool, Warm}\} \times \{\text{Same, Change}\}\}$

⇒ Angabe, ob ein Tag mit solchen Attributen zu dem Zielkonzept gehört ($f(x)$) also **positive** und **negative Beispiele** für Zielfunktion ($f(x)=0$ oder $f(x)=1$)

$\langle x^1, f(x^1) \rangle, \langle x^2, f(x^2) \rangle, \langle x^3, f(x^3) \rangle, \langle x^4, f(x^4) \rangle, \langle x^5, f(x^5) \rangle, \dots$

➤ Gesucht:

⇒ Hypothese h als Konjunktion von Attributen (z.B. $\langle ?, \text{Cold, High, }, ?, ?, ? \rangle$)

⇒ h ist Annäherung für „verborgene“ Zielfunktion $f \equiv \text{EnjoySport}: X \rightarrow \{0,1\}$

⇒ d.h.: Hypothese $h \in H$ so dass $h(x) = f(x)$ für alle $x \in D$

⇒ Solche h nennt man **konsistent** mit der Trainingsmenge D

➤ Trainingsannahmen:

⇒ es fehlen keine Werte

⇒ kein Rauschen in den Daten (widersprüchliche Daten)

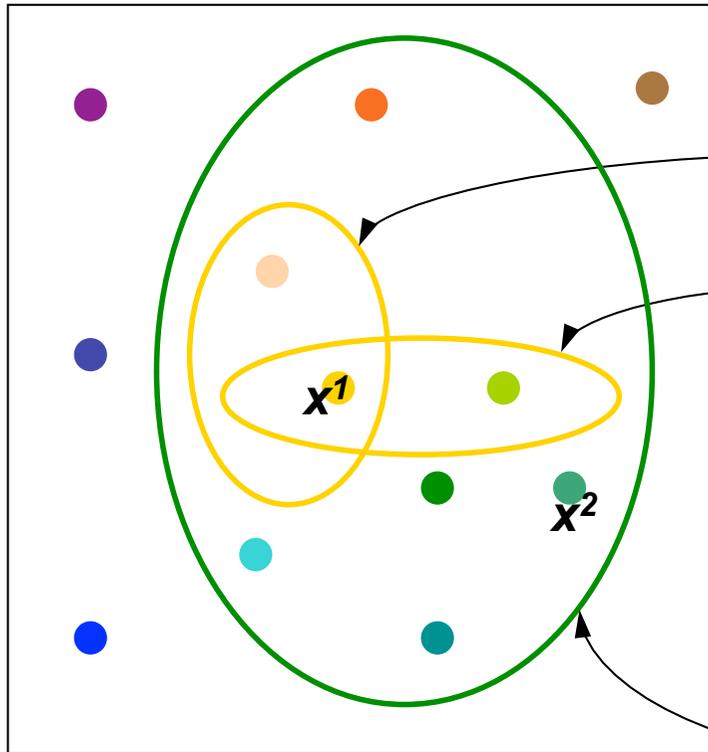
⇒ es gibt eine Hypothese, die konsistent mit D ist (die f annähert)

Grundidee des induktiven Lernens

- Induktion vs. Deduktion
 - ⇒ Deduktion: aus gegebenen Fakten neue Fakten ableiten (Beweis)
 - ✧ Gegeben: $A \Rightarrow B$, $B \Rightarrow C$, Folgerung: $A \Rightarrow C$
 - ✧ “Vom Allgemeinen zum Speziellen”
 - ⇒ Induktion: aus gegebenen Fakten neue Fakten mutmaßen (Hypothese)
 - ✧ Gegeben: Fliegt(Amsel), Fliegt(Storch), Folgerung: Fliegt(Vogel)
 - ✧ “Vom Speziellen zum Allgemeinen”
- Idee des induktiven Lernens:
 - ⇒ Lerne eine Funktion aus Beispielen durch (schrittweise) Verallgemeinerung
- Annahme beim induktiven Lernen
 - ⇒ Eine Hypothese, die für eine hinreichend große Beispielmenge die Zielfunktion gut annähert, wird die Funktion auch gut für unbekannte Beispiele annähern.
- Aber zunächst: Wie kann ein solches Lernverfahren konkret aussehen?

Instanzen- und Hypothesenraum

Instanzen X

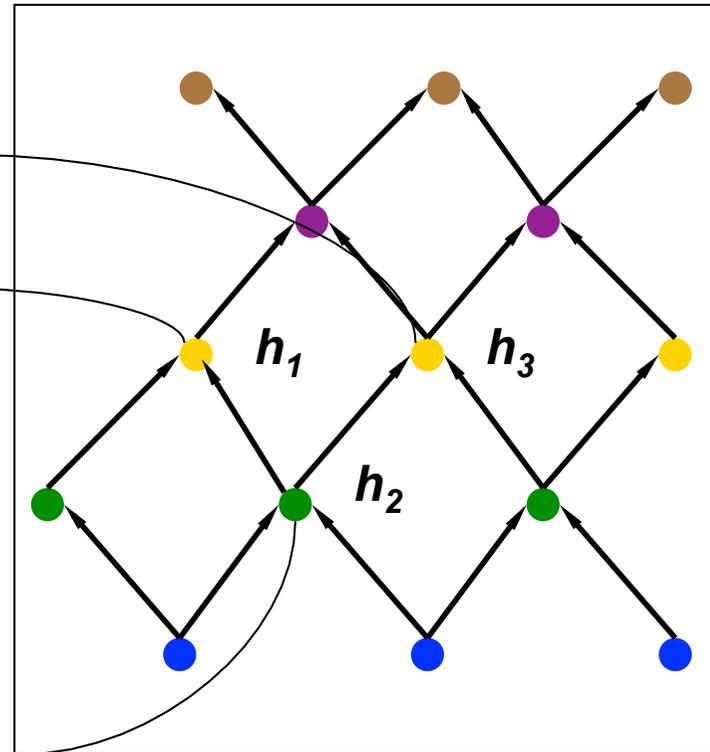


$x_1 = \langle \text{Sunny, Warm, High, Strong, Cool, Same} \rangle$
 $x_2 = \langle \text{Sunny, Warm, High, Light, Warm, Same} \rangle$

$\geq_g \equiv \text{More-General-Than-Or-Equal}$
 $>_g \equiv \text{More-General-Than}$

\geq_g ist partielle Ordnung:
 transitiv, reflexiv, antisymmetrisch

Hypothesen H , *partiell geordnet*



$h_1 = \langle \text{Sunny, ?, ?, Strong, ?, ?} \rangle$
 $h_2 = \langle \text{Sunny, ?, ?, ?, ?, ?} \rangle$
 $h_3 = \langle \text{Sunny, ?, ?, ?, Cool, ?} \rangle$

$h_2 \geq_g h_1, h_2 \geq_g h_3$

Aber weder $h_1 \geq_g h_3$ noch $h_3 \geq_g h_1$

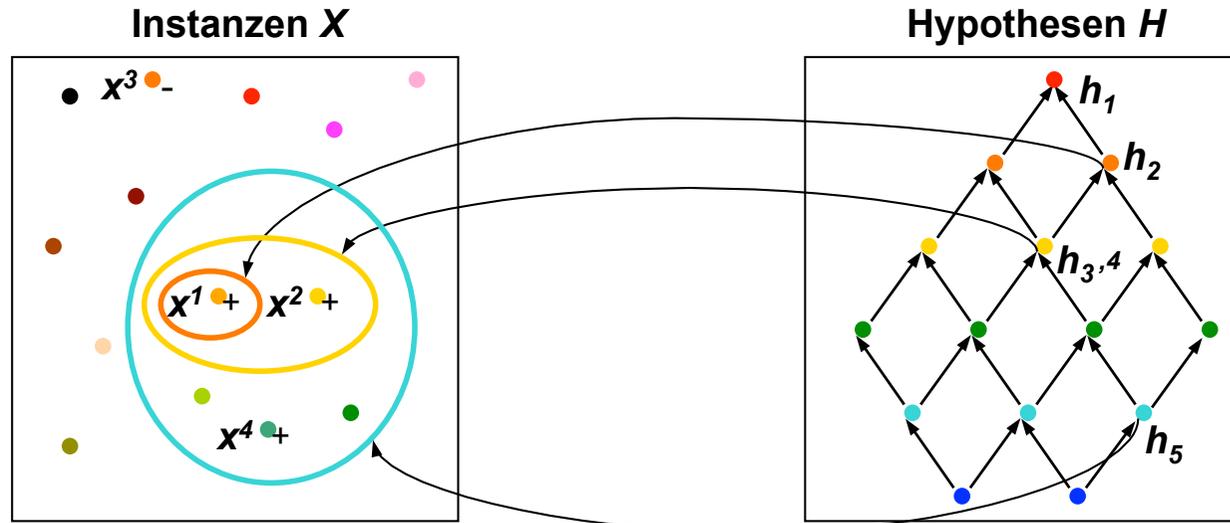
↑ Spezifisch
 ↓ Generell

Der Find-S Algorithmus

1. Initialisiere h als spezifischste mögliche Hypothese aus H
(In unserem Fall ist das die Hypothese $\langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$)
2. **FÜR ALLE** positive Instanzen x aus D , d.h. ($f(x)=1$)
FÜR ALLE Attribut mit Wert a_i in h
WENN a_i aus h wird durch x erfüllt
DANN keine Änderung
SONST ersetze a_i durch den nächst
allgemeineren Wert, der durch x erfüllt wird
3. Ausgabe von Hypothese h

- Bemerkungen:
 - ⇒ negative Beispiele werden nicht betrachtet
 - ⇒ Verfahren sucht "minimale" bzw. spezifischste Hypothese aus H , die alle positiven Beispiele umfaßt
 - ⇒ Wenn es eine Lösung gibt, dann wird sie auch gefunden
 - ⇒ Was ist, wenn es keine Lösung gibt?

Find-S: Suche im Hypothesenraum



$x_1 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle, +$
 $x_2 = \langle \text{Sunny, Warm, High, Strong, Warm, Same} \rangle, +$
 $x_3 = \langle \text{Rainy, Cold, High, Strong, Warm, Change} \rangle, -$
 $x_4 = \langle \text{Sunny, Warm, High, Strong, Cool, Change} \rangle, +$

$h_1 = \langle \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset \rangle$
 $h_2 = \langle \text{Sunny, Warm, Normal, Strong, Warm, Same} \rangle$
 $h_3 = \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$
 $h_4 = \langle \text{Sunny, Warm, ?, Strong, Warm, Same} \rangle$
 $h_5 = \langle \text{Sunny, Warm, ?, Strong, ?, ?} \rangle$

➤ Probleme von Find-S

- ⇒ Sagt nicht, ob der Begriff tatsächlich gelernt wurde
- ⇒ Keine Aussage, ob Daten inkonsistent sind
- ⇒ Spezifischste Hypothese muss nicht die einzige Lösung sein
- ⇒ Spezifischste Hypothese muss nicht die beste Lösung sein

Versionsraum

- Definition: konsistente Hypothesen
 - ⇒ Eine Hypothese h ist **konsistent** mit einer Menge von Trainingsdaten D eines Zielbegriffs f genau dann wenn $h(x) = f(x)$ für alle Beispiele $\langle x, f(x) \rangle$ aus D gilt
 - ⇒ $\text{konsistent}(h, D) := \forall \langle x, f(x) \rangle \in D: h(x) = f(x)$

- Definition: Versionsraum
 - ⇒ Der **Versionsraum** $VS_{H,D}$, zu einem Hypothesenraum H und Trainingsbeispielen D ist die Teilmenge der Hypothesen aus H , die konsistent mit allen Trainingsdaten aus D sind
 - ⇒ $VS_{H,D} := \{ h \in H \mid \text{konsistent}(h, D) \}$

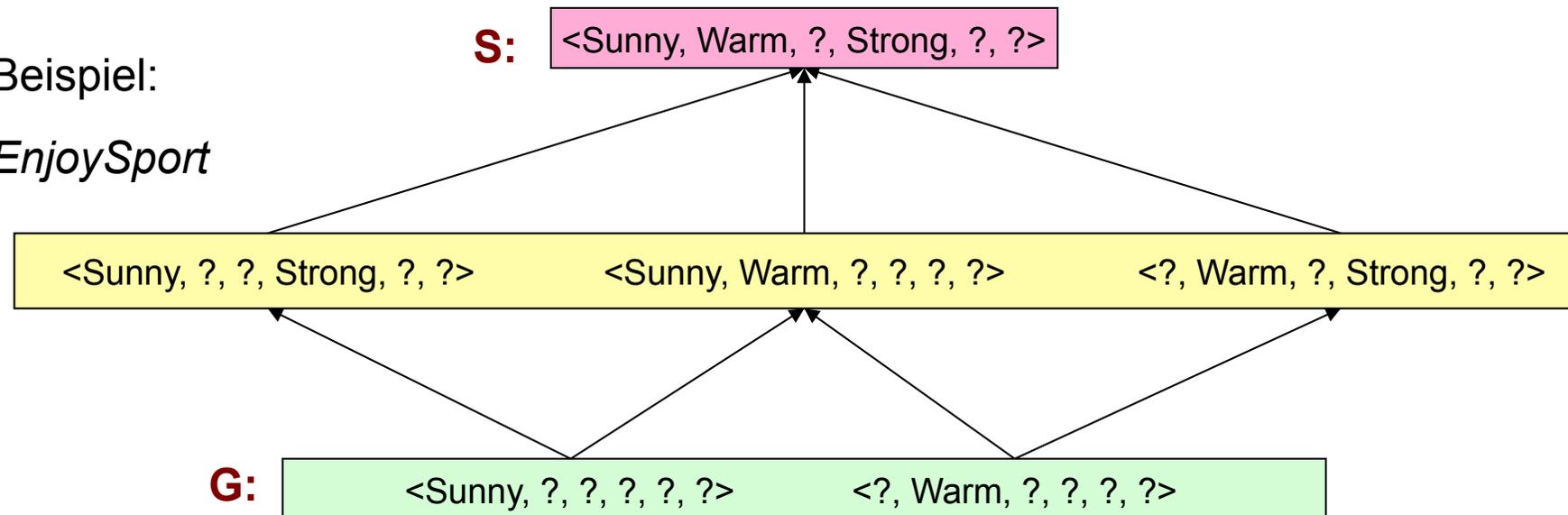
- Bemerkung:
 - ⇒ Wenn wir nicht nur an einer gültigen Lösung interessiert sind, dann ist der Versionsraum interessant.

Der List-Then-Eliminate Algorithmus

1. Initialisierung: **SETZE** $VS := H$,
 also ist VS die Liste aller Hypothesen
2. **FÜR ALLE** Trainingsbeispiele $\langle x, f(x) \rangle$
FÜR ALLE Versionen $h \in VS$
FALLS $h(x) \neq f(x)$
SETZE $VS := VS \setminus \{h\}$
3. Ausgabe des Versionsraums VS

Beispiel:

EnjoySport



Repräsentierung des Versionsraumes

➤ Bemerkung zum Versionsraum

- ⇒ Offensichtlich gibt es am meisten spezifische und am meisten generelle Versionen
- ⇒ Jede Hypothese, die “dazwischen” liegt, muss auch eine Lösung sein

➤ Definition: Generelle Schranke

- ⇒ Die generelle Schranke G des Versionsraumes $VS_{H,D}$ ist die Menge der generellsten Hypothesen
- ⇒ $G = \{g \in H \mid \text{konsistent}(g,D) \text{ und } \forall h \in H: (h >_g g \Rightarrow \neg \text{konsistent}(h,D))\}$
- ⇒ Das heißt, jedes h , das mehr Elemente zum Begriff dazunimmt als ein g , nimmt auch Fehler auf

➤ Definition: Spezifische Schranke

- ⇒ Die spezifische Schranke S des Versionsraumes $VS_{H,D}$ ist die Menge der spezifischsten Hypothesen
- ⇒ $G = \{g \in H \mid \text{konsistent}(g,D) \text{ und } \forall h \in H: (g >_g h \Rightarrow \neg \text{konsistent}(h,D))\}$
- ⇒ Das heißt, jedes h , das weniger Elemente im Begriff hat als ein g , dem fehlen gültige Beispiele

Versionsraum-Theorem

➤ Versionsraumtheorem

⇒ Jede Hypothese des Versionsraums liegt zwischen S und G

⇒ $VS_{H,D} = \{ h \in H \mid \exists s \in S: \exists g \in G: g \geq_g h \geq_g s \}$

➤ Bemerkung

⇒ Beweis: Übung und/oder bei Mitchell nachschauen

⇒ Hinweis:

✧ Zeige erst: Wenn $g, s \in VS_{H,D}$ und $g \geq_g h \geq_g s$, dann ist auch $h \in VS_{H,D}$

✧ Zeige dann: Wenn $h \in VS_{H,D}$, dann gibt es ein $g \in G$, mit $g \geq_g h$

- Beweisidee:

- Aus Definition von G folgt:

Es gilt entweder $h \in G$ oder es gibt ein $h' >_g h$, $h' \in VS_{H,D}$

- Dann kann man das gleiche mit h' fortführen.

- Da H endlich ist, muß man irgendwann an ein h'' kommen mit $h'' \in G$

✧ Und analog: Wenn $h \in VS_{H,D}$, dann gibt es ein $s \in S$, mit $h \geq_g s$

Kandidateneliminationalgorithmus

1. Initialisierung

$G = \{ \langle ?, \dots, ? \rangle \}$, Menge mit generellstem Element

$S = \{ \langle \emptyset, \dots, \emptyset \rangle \}$, Menge mit speziellstem Element

2. **FÜRALLE** Trainingsbeispiele $d = \langle x, f(x) \rangle \in D$

WENN $f(x) = 1$, d.h. für positive Beispiele

FÜRALLE $g \in G$ mit $g(x) = 0$ **SETZE** $G := G \setminus \{g\}$ (d.h. lösche alle $g \in G$, die schon zu klein sind)

FÜRALLE $s \in S$ mit $s(x) = 0$

SETZE $S := S \setminus \{s\}$

Erweitere S um alle minimalen Generalisierungen h von s , so dass

- $h(x) = 1$, d.h., h ist konsistent mit d

- $\exists g \in G: g \geq_g h$, d.h., h liegt noch unter der generellen Schranke

Lösche alle $s \in S$, die genereller sind als andere Elemente aus S

WENN $f(x) = 0$, d.h. für negative Beispiele

FÜRALLE $s \in S$ mit $s(x) = 1$ **SETZE** $S := S \setminus \{s\}$ (d.h. lösche alle $s \in S$, die schon zu groß sind)

FÜRALLE $g \in G$ mit $g(x) = 1$

SETZE $G := G \setminus \{g\}$

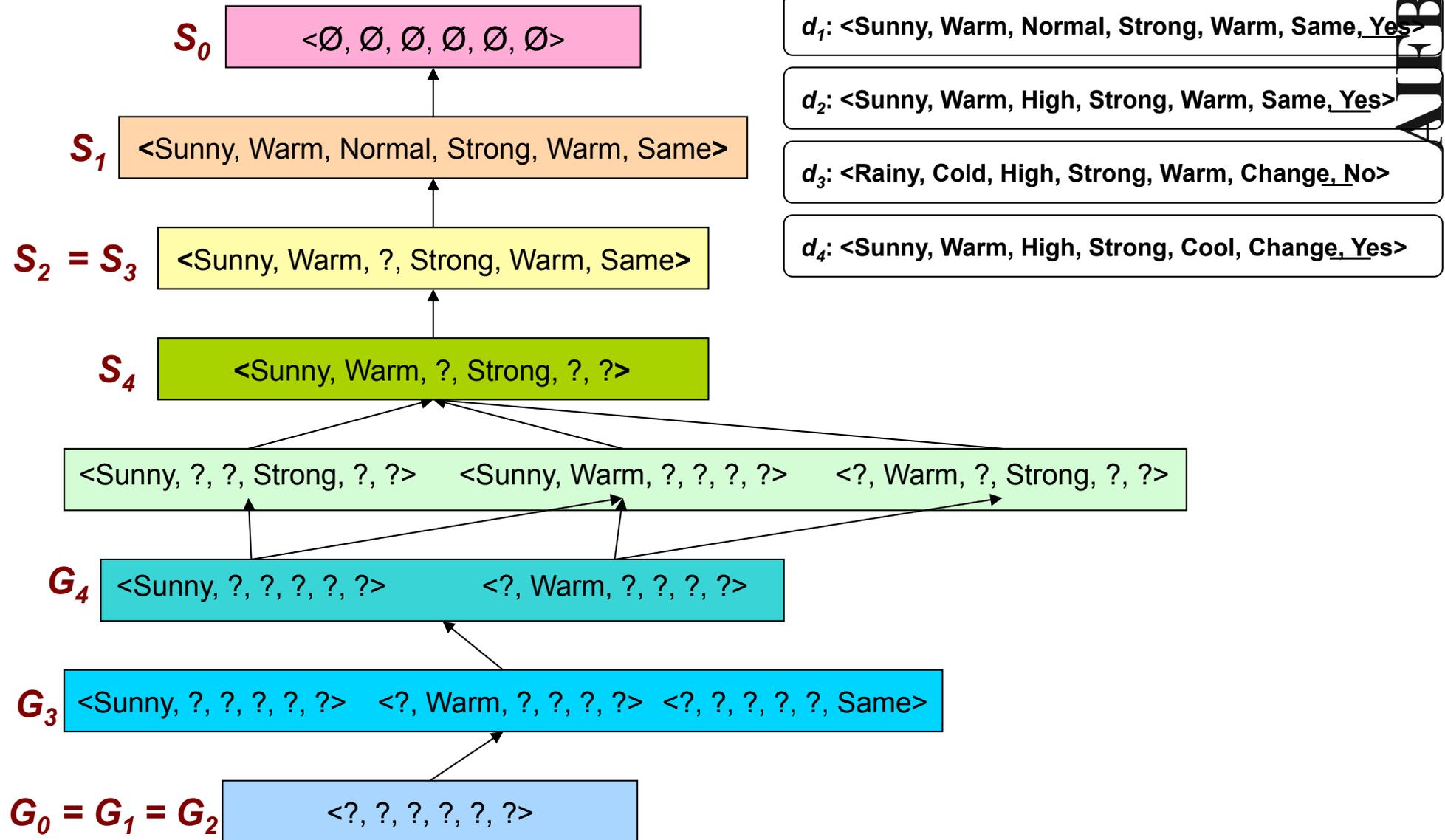
Erweitere G um alle minimalen Spezialisierungen h von g , so dass

$h(x) = 0$, d.h., h ist konsistent mit d

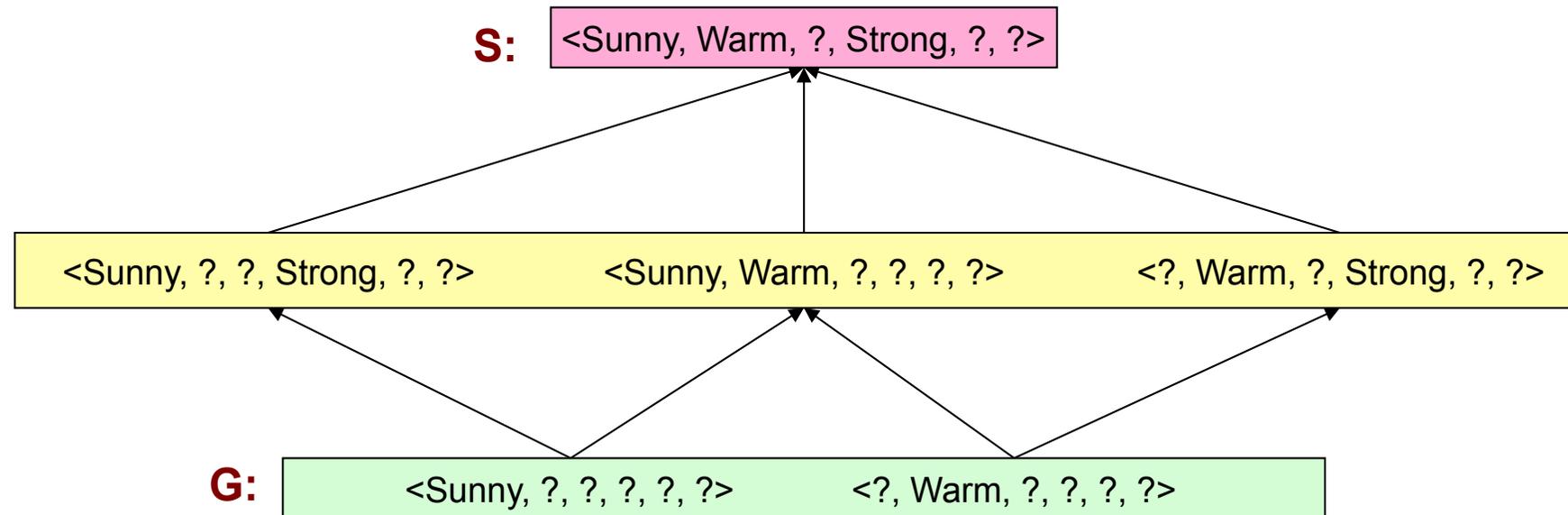
$\exists s \in S: h \geq_g s$, d.h., h liegt noch über der speziellen Schranke

Lösche alle $g \in G$, die spezieller sind als andere Elemente aus G

Beispiellauf



Was macht man mit dem Versionsraum?



- Bewertung von neuen Beispielen
 - ⇒ <Sunny, Warm, Normal, Strong, Cool, Change>
 - ⇒ <Rainy, Cold, Normal, Light, Warm, Same>
 - ⇒ <Sunny, Warm, Normal, Light, Warm, Same>
 - ⇒ <Sunny, Cold, Normal, Light, Warm, Same>
- Möglichkeiten:
 - ⇒ Klare Entscheidungen, Mehrheiten, Unentschieden
- Auswahl neuer Trainingsbeispiele
 - ⇒ Ideal: solche, die den Versionsraum in zwei gleich große Teile teilen

Generalisierung beim induktiven Lernen

➤ Beispiel für induktive Generalisierung

⇒ Positive Beispiele:

<Sunny, Warm, Normal, Strong, Cool, Change, Yes>

<Sunny, Warm, Normal, Light, Warm, Same, Yes>

⇒ Gefolgertes S:

<Sunny, Warm, Normal, ?, ?, ?>

➤ Fragen:

⇒ Was rechtfertigt eine Generalisierung?

✧ z.B. <Sunny, Warm, Normal, Strong, Warm, Same>

⇒ Wann gibt es ausreichende Informationen für eine Generalisierung?

Der induktive Bias

- Bias (engl.) – Vorliebe, Voreingenommenheit, Befangenheit, systematischer Fehler, ...
- Induktiver Bias:
 - ⇒ Menge der möglichen Hypothesen beschränkt die möglichen Lösungen h , die f annähern können
 - ⇒ Der induktive Bias beschreibt alle Grundannahmen, die in dem Lern- und Klassifikationsverfahren stecken.
 - ⇒ Ohne induktiven Bias gibt es keine Generalisierung!

Lerner mit unterschiedlichem Bias

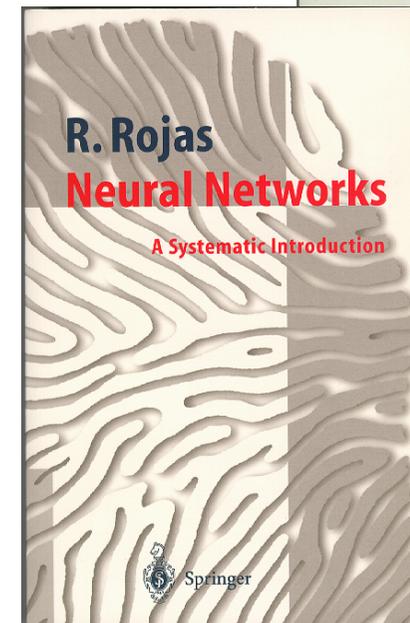
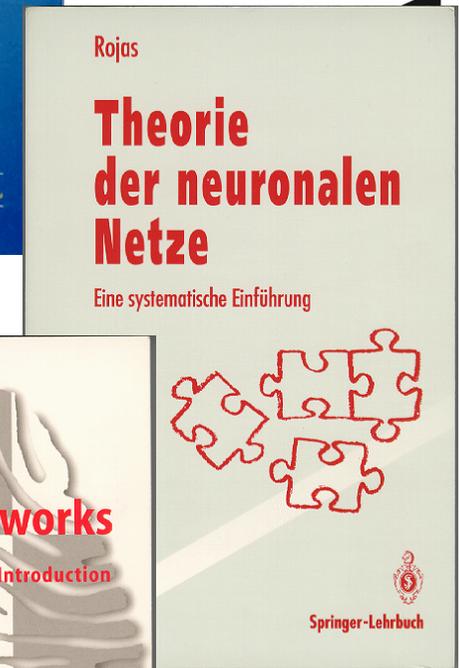
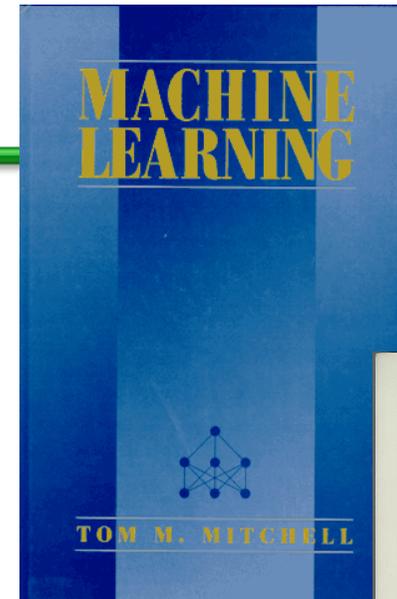
- Lerner ohne Bias, “Auswendiglernen”
 - ⇒ Nur Klassifizierung von vorher Gesehenem
 - ⇒ Speichert Beispiele
 - ⇒ Kann nur solche x klassifizieren, die schon gesehen wurden
- Versionsraum und Kandidateneliminationsalgorithmus
 - ⇒ Stärkerer Bias: Konzepte lassen sich als h aus H beschreiben
 - ⇒ Speichert Schranken für Generalisierungen und Spezialisierungen
 - ⇒ Klassifikation von x genau dann, wenn es im Versionsraum liegt und alle Versionen in der Bewertung übereinstimmen
- Kandidateneliminationsalgorithmus mit Mehrheitsentscheidung
 - ⇒ Noch stärkerer Bias: Konzepte lassen sich als h aus H beschreiben
 Mehrheit der Konzepte in H wird recht haben
 - ⇒ Klassifikation von x , wenn es eine mehrheitliche Klassifikation der Versionen gibt
 (z.B. mehr als 70%)
 - ⇒ Hier sind implizit statistische Annahmen vorhanden, wie repräsentativ die Trainingsbeispiele sind
- Find-S
 - ⇒ Noch stärkerer Bias: die spezifischste Hypothese gilt
 - ⇒ Implizite Annahme: alle noch nicht bekannten Beispiele sind negativ
 - ⇒ Klassifiziert x anhand von S

Zusammenfassung

- Begriffslernen ist eine Suche in H
 - ⇒ Hypothesenraum H ist der Suchraum
 - ⇒ Lernen entspricht Suchen und Finden der richtigen Hypothese
- Hypothesenraum kann geordnet werden (generell-spezifisch)
 - ⇒ Die more-general-than Relation ist eine partielle Ordnung
 - ⇒ H hat eine obere und untere Schranke
- Versionsraum und Kandidateneleminationsalgorithmus
 - ⇒ S und G Schranken beschreiben die Unsicherheit der Lerner
 - ⇒ Versionsraum kann genutzt werden, um Vorhersagen für neue Beispiele zu machen
- Verfahren kann genutzt werden, um Beispiele auszuwählen
- Jedes Beispiel muss nur einmal gezeigt werden
- Induktiver Bias
 - ⇒ zusätzliche Annahmen über Struktur der Hypothesen erhöhen die Generalisierungsfähigkeit
 - ⇒ Lernen mit unterschiedlich hohem Bias

Literatur

- Machine Learning, Tom Mitchell, McGraw Hill, 1997
- "Neural Networks - A Systematic Introduction", Raul Rojas, Springer-Verlag, Berlin, 1996.
- "Theorie der neuronalen Netze", Raul Rojas, Springer-Verlag, Berlin, 1993/99.
- ...



Weitere Lernverfahren

➤ Symbolisch

- ⇒ Entscheidungsbäume
- ⇒ Fallbasiertes Schließen (CBR)
- ⇒ ...

➤ Subsymbolisch

- ⇒ Neuronale Netze
- ⇒ Support Vector Machines
- ⇒ Genetische Algorithmen
- ⇒ ...