

① Maschinelles Lernen 1 Deduktives Lernen (Topic 11)

Deduktion Eigenschaften

„Deduktion“ = „Ableitung“

- Benutzung von vorhandenem Wissen, um daraus auf neues Wissen zu schließen (vom Allgemeinen zum Speziellen)
- { - gelerntes Wissen ist korrekt (gegeben ursprüngliches Wissen korrekt) \oplus
- wenige oder gar keine Beispiele benötigt \oplus
- kein induktiver Bias \oplus
- { - nicht empirisch, sondern axiomatisch
- { - Ausnutzung vorhandenen domänen spezifischen Wissens \ominus

Beispiel Schach:

Situation: König und andere weiträumige Figur gleichzeitig bedroht

\rightarrow Suchraum ist zu groß

\rightarrow induktives Verfahren braucht sehr viele Beispiele

\rightarrow Mensch (deduktiv) ist effizienter! Wenige Bsp.! reichen

Das extrahierte Wissen: wenn König und Turm bedroht \rightarrow bewege König, verliere Turm \Rightarrow möglich mit „Bereichstheorie“

Erläuterungsbasiertes Lernen

- ermöglicht das Erzeugen einer, den Anforderungen gerechten, Generalisierung aus einem einzigen positiven Beispiel.

Gegeben: 1) Zielkonzept: Beschreibung des zu erlernenden Konzepts
2) Trainingsbeispiel (für das Zielkonzept)
(„Domain Theory“) 3) Bereichstheorie: Regeln und Fakten, die erklären, warum das Trainingsbeispiel ein Bsp. für Zielkonzept ist
4) Operationalitätskriterium: Prädikat, das die Form spezifiziert, in der erlernte Beschreibungen vorliegen müssen

Gesucht: eine Generalisierung des Trainingsbeispiels, die die Definition des Zielkonzepts und das Operationalitätskriterium erfüllt

\Rightarrow Ziel: Beispiel \leftrightarrow Generalisierung

Erläuterungsbasierte Generalisierung (EBG)

→ Zweischritt-Verfahren für jedes positive Beispiel

• Explain (Finden einer Erläuterung)

• Generalize (Generalisierung der Erläuterung)

EBG - Beispiel: Zielkonzept \rightarrow robust(x) (ist Roboter robust: ja/nein)
Trainingsbeispiele \rightarrow robot(Num5), r2d2(Num5), agd(Num5), manufacturer(Num5, GR)...

Bereichstheorie:

1) fixes(u,u) \rightarrow robust(u)

Ein Individuum, dass sich selbst reparieren kann, ist robust.

2) sees(x,y) \wedge habile(x) \rightarrow fixes(x,y)

Ein gerüstetes Ind. das eine andere Entität sieht, kann diese reparieren.

3) robot(w) \rightarrow sees(w,w)

Alle Roboter können sich selbst sehen.

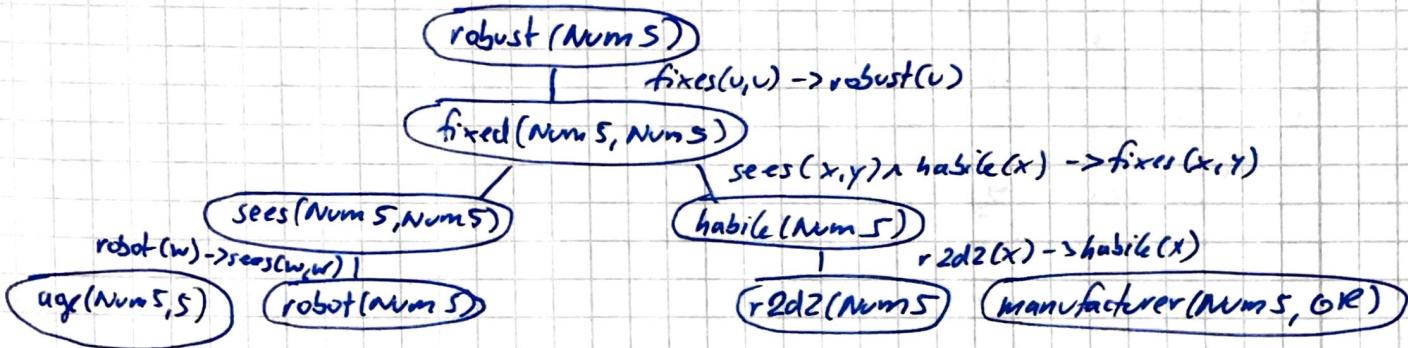
1) $r2d2(x) \rightarrow habile(x)$

R2D2-Klassenindividuen sind geschickt.

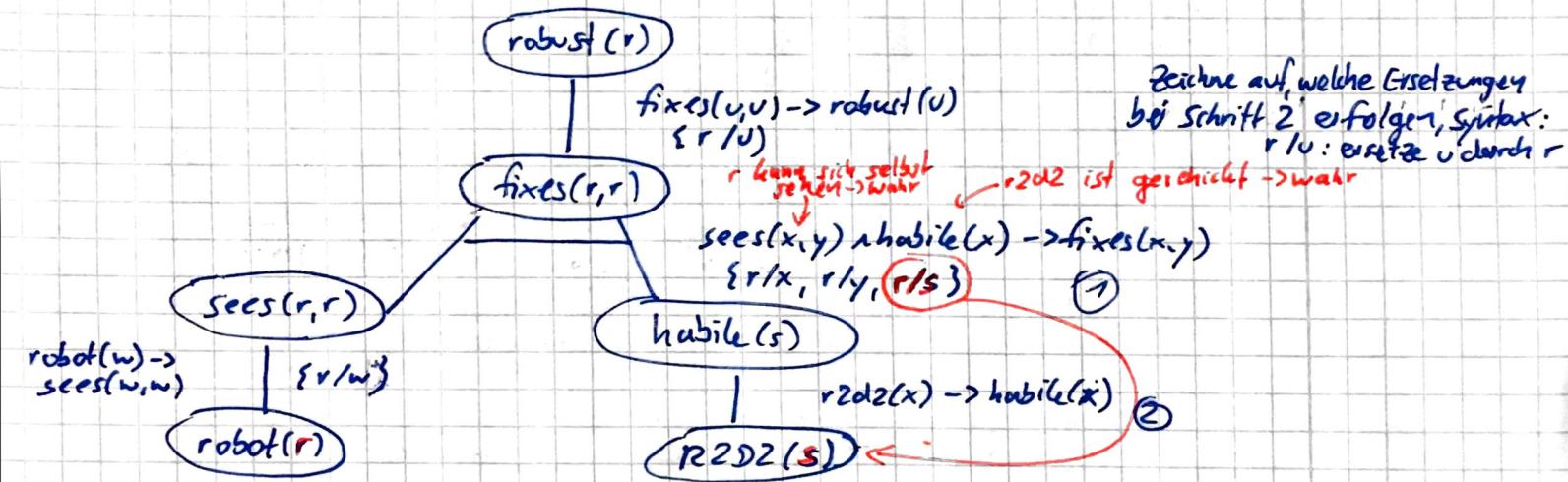
(2)

Operationalitätskriterium: Das Zielkonzept ist beschrieben in Termen der Beispielbeschreibung, z.B. $hallo(x, y)$

1. Schritt: Explain \Rightarrow Konstruiere in Termen der Bereichstheorie eine Erklärung, die zeigt, wie das Trainingsexample das Zielkonzept erfüllt, brachte dabei, dass Erklärung in einem Ausdruck endet, die das Operationalitätsh. erfüllt



2. Schritt: Generalize \Rightarrow hinreichende Bedingungen bestimmen, unter denen die Erklärung gültig ist, formulierte Terme so, dass Operationalitätsh. erfüllt



Generalisiere durch Ersetzen der Konstanten Num 5 durch eine Variable. Alternativ (wie hier): nutze zwei Variablen um manchmal noch stärker zu generalisieren: r und s . An den Stellen ① beim „Hochwandern“ des Baums fällt auf, dass aus Variablen r und s der gemeinsame „Menn“ r hervortritt. Wende in ② nur aufgezählte Ersetzungen an \rightarrow neue Regel zeigt sich:

$$\underline{\underline{robot(r) \wedge r2d2(r) \rightarrow robust(r)}}$$

\Rightarrow füge zu Bereichstheorie hinzu

Was ist erklärbungsbasierter Lernen?

- Prozess, der implizites Wissen in explizites umwandelt
- für jedes positive Trainingsbeispiel wird eine Generalisierung erstellt

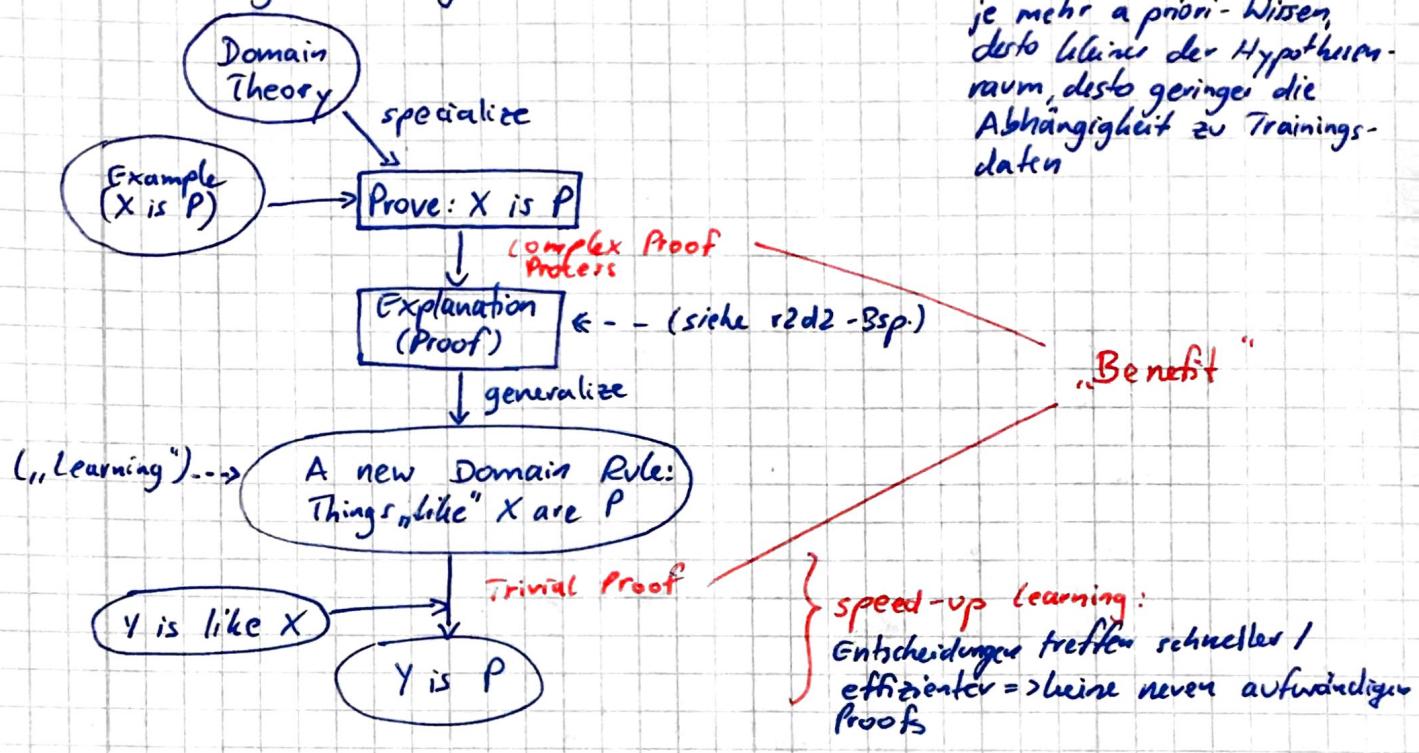
Warum erklärbungsbasiertes Lernen?

- Analogien zur menschlichen Wissensverarbeitung
- kein Bias wie induktivem Lernen

Wichtig: Speedup-Learning

(3) Maschinelles Lernen 1 Deduktives Lernen (Topic 11)

Erläuterungsbasierte Generalisierung Schaubild



Probleme bei EBL:

- 1) Erläuterungen (Explanation) nicht möglich
- 2) Inkonsistente (Bsp. + Gegenbeispiel) oder multiple Erläuterungen

Lösungen: Approximierungen (Annahme, etwas sei wahr/falsch), Explorierung

Frage 1: EBL-System kann Bsp. erklären. Kann es auch Bsp. erzeugen?
Nicht gegeben, denn es gilt nicht immer:

Erläuterungsfähigkeit => Generierungsfähigkeit
(Bsp.: NP-vollständige Probleme)

→ Bsp. dienen als Richtungszeiger → Hypothesenraum kleiner

Frage 2: Wird bei EBL tatsächlich gelernt? Oder wird Wissen erlernt, das bereits im Hintergrundwissen enthalten ist?

Nach Definition: ein System lernt aus Erfahrung, wenn seine Leistungen bei seinen Aufgaben mit Messungen nachweislich durch Erfahrung steigt.

*ded. Hülle: die Menge aller Aussagen, die aus den gegebenen hergeleitet werden können

EBL/EBG verändert die "deduktive Hülle" der Wissensbasis nicht.
=> unbegrenzte Ressourcen vorausgesetzt wird nichts neues gelernt

Aber: in Praxis begrenzte Ressourcen

Daher: es wird tatsächlich gelernt → Speed-up learning:
Entscheidungen sind effizienter getroffen. Speed-up learning erzeugt kein System, das Entscheidungen treffen kann, die es nicht vor dem Lernen hätte auch treffen können.

Frage 3: Wie kann Ressourceneinsatz für das Lernen neuer Regeln reduziert werden?
① → Beschränkung der Aufnahme neuer Regeln
→ Messen der Nützlichkeit neuer Regeln
→ Verwerfen/Erhalten der Regel
② → Paralleles Rechnen (Matching)

Anwendungen von EBL

• Erzeugung von Makrooperatoren

(4)

-> Stanford Research Institute Problem solver - STRIPS

-> Symbolic Cognitive Architecture - SOAR

• Lernen, Sich zu kontrollieren

-> PRODIGY (Effektivitätssteigerung beim Planen bei großen Zustandsräumen)

Deduktives Lernen vs. Induktives Lernen

IL

Ziel: Hyp. passt zu den Daten

DL

Hyp. passt zur Bereichstheorie

Vorteile: wenig a-priori - Wissen

wenig Beispiele notwendig

Nachteile

- schlecht bei geringen Datenmengen
- schlecht bei inkorrektem Bias

schlecht, falls schlechte Bereichstheorie

Induktives Lernen

- viele Daten
- wenig Vorwissen

reale Lernprobleme

Deduktives Lernen

- wenig Daten
- großes Vorwissen

(durch Statistik gerechtfertigte Hyp.)

(durch Logik gerechtfertigte Hyp.)

=> Reale Lernprobleme bewegen sich zwischen induktiv und deduktiv

- > a-priori-Wissen vorhanden aber nicht beliebig viel
- > ist a-priori-Wissen korrekt?
- > Trainingsdaten begrenzt in der Menge
- > sind Trainingsdaten korrekt?

Anforderung an hybride Verfahren

- mit unvollständigem a-priori - Wissen und Daten besser sein als rein induktive und rein deduktive Verfahren
- ohne a-priori - Wissen so effektiv lernen wie rein induktive Verfahren
- etc. ...

Definition des Lernproblems bei hybriden Verfahren

Gegeben:

- Trainingsdaten D , evtl. mit Fehlern
- Bereichstheorie B , evtl. mit Fehlern
- Hypothesenraum H

Ziel: Finde Hyp. h , die am besten sowohl zu D als auch zu B passt
Entscheidung über beste Hyp. z.B. mit:

$$\arg \min_{h \in H} k_D \bar{E}_D(h) + k_B \bar{E}_B(h)$$

\uparrow
 Fehlerrate
 Daten

\uparrow
 Fehlerrate
 Bereichstheorie

k_D Gewicht Daten
 k_B Gewicht Bereichstheorie

Frage: wie k_D und k_B wählen?

5. Maschinelles Lernen \neq Deduktives Lernen (Topik 11)

Ausätze zur Konstruktion hybrider Verfahren

- H Hypothosenraum
- h_0 initiale Hypothese
- O Suchoperatoren
- G Ziellkriterium

Eine Möglichkeit: verwende Vorwissen,

- > um initiale Hyp. h_0 abzuleiten $\Rightarrow \text{KBANN}$
- > um Ziellkriterium G der Suche zu ändern
- > um mögliche Suchschritte zu ändern

KBANN

- Idee
- initialisiere NN mittels Bereichstheorie
 - verfeinere initiales Netz durch Backprop und Trainingsexempel
 - => Wenn Bereichstheorie korrekt:
alle Bsp. korrekt klassifiziert
 - => Wenn nicht korrekt:
Fehler in Bereichstheorie, induktive Verfeinerung durch Backprop
- Intuition: sogar wenn Bereichstheorie nur annähernd korrekt, Starthypothese besser als zufällige Initialisierung

Details

Gegeben: Menge von Trainingsbeispielen + Bereichstheorie (log. Forme)

Geucht: NN, das zu Daten paust, angelehnt an Bereichstheorie

Algorithmus

1. Pro Instanzattribut ein Eingangsneuron
2. Pro Klausel der Bereichstheorie ein Neuron wie folgt einfügen:
 - > für jedes Attribut eine Verbindung zu Klausel-Neuron
 - > für nicht-negiertes Attribut: verwende positives Gewicht W
 - sonst: negatives Gewicht -W
 - > Setze Schwellwert auf $-(n-0.5)W$
(mit n = Anzahl der nicht-negierten Bedingungssteile)
3. Zusätzliche Verbindungen, um jedes Neuron auf Schicht i mit jedem auf Schicht $i+1$ zu verbinden, zufällige kleine Gewichte zuweisen
4. Backprop mit Trainingsdaten anwenden

KBANN-Anwendungen

- Lernen von physikalischen Objektklassen
 - z.B. Tasse (Instanzen sind Angaben über Material)
- Erkennung biologischer Konzepte in DNS-Sequenzen
 - Gen-Sequenzen repräsentiert als Strings (Daten)
 - geucht: bestimmte genetische Regionen Bereichstheorie aus biol. Forschung