

Inductive Rule Learning on the Knowledge Level

Induktives Lernen von produktiven Regelmengen auf der Wissensebene

Ute Schmid

(inspiriert von Fritz Wysotzki
gemeinsame Arbeit mit

Emanuel Kitzelmann, Martin Hofmann und anderen)

Kognitive Systeme, Fakultät Wirtschaftsinformatik und Angewandte Informatik
Otto-Friedrich Universität Bamberg
<http://www.uni-bamberg.de/kogsys>



2th LEIBNIZ CONFERENCE OF ADVANCED SCIENCE

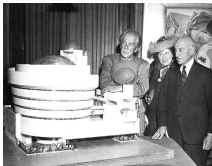


Berlin, 7.12.2011



Expertise

*An expert is a man who has stopped thinking –
he knows!* (Frank Lloyd Wright)



Frank Lloyd Wright, Hilla Rebay, Solomon Guggenheim at the unveiling of the model for the Guggenheim Museum, August, 1945
www.westporthistory.org

Expertise

*An expert is a man who has stopped thinking –
he knows!* (Frank Lloyd Wright)



Frank Lloyd Wright, Hilla Rebay, Solomon Guggenheim at the unveiling of the model for the Guggenheim Museum, August, 1945
www.westporthistory.org

- *gut feeling*: automatisierte, schnelle Klassifikation
- Schnelle, fehlerfreie, optimale Generierung von Problemlösungen:
Anwendung komplexer Handlungsregeln
- **Induktiver Erwerb von Wissen aus Erfahrung**

Induktives Lernen

- Menschliche Fähigkeit, komplexe Anforderungen zu bewältigen: basiert in großem Maße auf der Fähigkeit, Erfahrungen zu nutzen:
 - ▶ Vorhersage von Eigenschaften/Reaktionen von (natürlichen/künstlichen, unbelebten/belebten) Objekten
 - ▶ Schlussfolgern über mögliche Ergebnisse von Handlungen
 - ▶ Anwendung von in der Vergangenheit erfolgreichen Routinen und Strategien auf neue Aufgaben und Probleme
- Erfahrung: häufig ein Strom (nur positiver) Beispiele (unüberwachtes Generalisieren)
- Grundlegender Prozess, um Wissen zu erweitern: **Induktive Inferenz**
Konstruktion von Hypothesen so, dass Wissen aus vorangegangenen Erfahrungen auf neue Situationen übertragen werden kann
- Philosophie (Goodman, 1965), KI (maschinelles Lernen), Psychologie (Klahr & Wallace, 1976; Tenenbaum, Griffiths, & Kemp, 2006)

Holland, Holyoak, Nisbett & Thagard (1986). Induction – Processes of inference, learning, and discovery. Cambridge, MA: MIT Press.

Induktives Lernen

Aus Perspektive der Informatik: Maschinelles Lernen

- Ansatz zur Wissensakquisition (statt direktem *knowledge engineering*)
- Möglichkeit, um adaptive, flexible Systeme zu entwickeln
- Viele erfolgreiche Ansätze, insbesondere seit den 90er Jahren: Entscheidungsbaumalgorithmen (ID3, C4.5), künstliche neuronale Netze, SVM, Bayes-Klassifikatoren, kNN, Reinforcement Learning, genetische/evolutionäre Ansätze, Inductive Logic Programming, ...

Aus Perspektive der Kognitionswissenschaft

- *How can a cognitive system process environmental input and stored knowledge so as to benefit from experience?* (Holland, Holyoak, Nisbett & Thagard, 1986, Induction – Processes of Inference, Learning, and Discovery)

Induktives Lernen

Wechselseitige Befruchtung von KI und Psychologie

- Erste Entscheidungsbaum-Algorithmen von Hunt, Marin and Stone (1966) angeregt durch Studien zum Konzepterwerb von Bruner, Goodnow & Austin (A Study of Thinking, 1956)
 - Parallele Entwicklungen: Exemplar-basierte Ansätze in der Psychologie (Nosofski), *lazy learning* (z.B. *k*NN) im maschinellen Lernen
-
- Lernen findet auf vielen Ebenen statt:
Basale Lernprozesse (auf Sensordaten) —
Symbolbasierte Ansätze (auf der Wissensebene)
 - Fokus: Lernen auf der Wissensebene, speziell **produktive Regeln**

Produktive Regelmengen

- Spezifischer Aspekt der induktiven Inferenz: Erwerb **produktiver Regelmengen** aus Erfahrung
- Chomsky (1965): Ein System von Regeln ist produktiv, wenn es auf Situationen unterschiedlicher Komplexität angewendet werden kann
 - ▶ Grammatikregeln (die aus der Mode gekommene Idee eines LAD)
 - ▶ Rekursive Konzepte
odd number, prime number, sorted list, ancestor
 - ▶ Erzeugung und Anwendung regulärer Handlungssequenzen
blocksworld, Tower of Hanoi
- Typischerweise: Erfahrung mit kleinen Mengen von Beispielen und Induktion einer unendlichen Regularität, die die Anwendung auf neue Beispiele beliebiger Komplexität erlaubt

Produktive Regelmengen

Eigenschaften

- **Kompetenz** (zugrundeliegendes Wissen) vs. Performanz (anfällig für unsystematische Variationen und Fehler)
- **Prozedurales/implizites** Wissen: Automatismen, die das Arbeitsgedächtnis kaum belasten
- **Verbalisierbares** Wissen: Ein rekursives Konzept oder eine Problemlösestrategie kann einer anderen Person erklärt werden
- In kognitiv technischen Systemen: symbolische Regeln, Erklärungskomponente

Induktive Programmsynthese als Zugang zum Lernen produktiver Regeln

Induktive Programmierung

- Anwendung von maschinellem Lernen auf das Problem der Programmsynthese
- Automatische Erzeugung (rekursiver) Programme aus Eingabe-/Ausgabe-Beispielen (unvollständigen Spezifikationen)
- Ansätze:
 - ▶ **analytisch, daten-getrieben:** THESYS, GOLEM, IGOR (ILP und induktive funktionale Ansätze)
 - ▶ **generate-and-test:** FFOIL (ILP), ADAPT (evolutionär), MAGICHASKELLER (systematische Aufzählung)

(Flener & Schmid, AI Review, 29(1), 2009; Encyclopedia of Machine Learning, 2010; Schmid, Kitzelmann & Plasmeijer, AAIP 2009; AAIP 2011)

E/A Beispiele:

last [a] = a

last [a,b] = b

last[a,b,c] = c

last[a,b,c,d] = d

Induziertes Programm:

last[x] = x

last(x:xs) = last(xs)

IGOR2

- Analytischer Ansatz zum Lernen funktionaler Programme (MAUDE, HASKELL)
- Effiziente Induktion rekursiver Regelmengen aus wenigen positiven Beispielen
- Lernt lineare, Baum-, und wechselseitig rekursive Regelmengen
- Automatische Induktion von Hilfsfunktionen (*necessary function invention*)
- Kann Hintergrundwissen berücksichtigen (analog zum ILP-System GOLEM)
- Restriction Bias: funktionale rekursive Programme, bei denen die äußere Funktion entweder nicht-rekursiv oder aus dem Hintergrundwissen ist
- Nicht-rekursive Programme zur Klassifikation als Spezialfall (z.B. playTennis, Mitchell, 1997)
- Präferenz-Bias: Weniger Fallunterscheidungen, spezifischere Patterns, weniger rekursive Aufrufe

(Kitzelmann and Schmid, JMLR 2006; Schmid et al., AGI 2009, Hofmann et al. KI 2008, Kitzelmann, AAIP 2009)

Empirische Ergebnisse

	<i>isort</i>	<i>reverse</i>	<i>weave</i>	<i>shiftr</i>	<i>mult/add</i>	<i>allodds</i>
ADATE	70.0	78.0	80.0	18.81	—	214.87
FLIP	×	—	134.24 [⊥]	448.55 [⊥]	×	×
FFOIL	×	—	0.4 [⊥]	< 0.1 [⊥]	8.1 [⊥]	0.1 [⊥]
GOLEM	0.714	—	0.66 [⊥]	0.298	—	0.016 [⊥]
IGOR II	0.105	0.103	0.200	0.127	⊙	⊙
MAGH.	0.01	0.08	⊙	157.32	—	×

	<i>lasts</i>	<i>last</i>	<i>member</i>	<i>oddeven</i>	<i>multlast</i>
ADATE	822.0	0.2	2.0	—	4.3
FLIP	×	0.020	17.868	0.130	448.90 [⊥]
FFOIL	0.7 [⊥]	0.1	0.1 [⊥]	< 0.1 [⊥]	< 0.1
GOLEM	1.062	< 0.001	0.033	—	< 0.001
IGOR II	5.695	0.007	0.152	0.019	0.023
MAGH.	19.43	0.01	⊙	—	0.30

— not tested × stack overflow ⊙ timeout ⊥ wrong

all runtimes in seconds



Evaluation

IGOR2 ...

- ist sehr effizient und ist mächtiger als andere analytische IP Ansätze
- verfügt über einen Mechanismus zur *necessary function invention*
- existiert in einer noch allgemeineren Variante, die auf der Identifikation von Merkmalen von Funktionen höherer Ordnung in den Beispielen basiert (*general fold*) (Doktorarbeit von Martin Hofmann, 2010)

Download via <http://www.uni-bamberg.de/kogsys/effalip>

IGOR2 als *Cognitive Rule Acquisition Device*

- Analytische induktive Programmierung bietet einen allgemeinen Mechanismus um generalisierte Regelmengen aus Beispielen für ein gewünschtes Verhalten zu extrahieren
- Typische Bereiche, in denen aus positiven Beispielen gelernt wird
 - ▶ Semantische Relationen (Ist-Ein, Vorfahr)
 - ▶ Sprache (regelmäßige Beugungen, grammatische Strukturen)
 - ▶ Problemlösen (Turm von Hanoi)
- Plötzliches Gefühl, ein Problem verstanden zu haben (Aha-Effekt): Regularitäten wurden zu einem (Problemlöse-) Schema generalisiert
- Randbemerkung: Ich postuliere nicht, dass entweder (a) symbolische rekursive Regelmengen im Gehirn gespeichert werden oder (b) Menschen bewusst mit dem kognitiven System zugänglichen rekursiven Regelmengen umgehen!
stattdessen: Rekursive Regelmengen sind eine elegante Möglichkeit, das entsprechende Wissen formal zu fassen

Lernen aus Problemlöseerfahrung

Anwendung von IP auf Lernen beim Problemlösen

- Idee: Lerne bei einem Problem mit geringer Komplexität und generalisiere die produktive Regelmenge, die Handlungssequenzen für Probleme aus dem gleichen Bereich mit beliebiger Komplexität generieren kann
- Beispiel: Erzeuge einen Plan für den Turm von Hanoi mit 3 Scheiben und generalisiere die Strategie für n Scheiben
- IGORII ist *ein*, sicher nicht der einzig mögliche, Zugang
Aber: datengetriebene Ansätze scheinen kognitiv plausibler als Erzeuge-und-Teste Ansätze

(Schmid, Hofmann, Kitzelmann, AGI'2009; Schmid & Wysotzki, AIPS 2000; Schmid, LNAI 2654; Schmid & Kitzelmann CSR, 2011)

Lernen der Strategie für Turm von Hanoi

Input to IGOR2

```
eq Hanoi(0, Src, Aux, Dst, S) =  
  move(0, Src, Dst, S) .  
eq Hanoi(s 0, Src, Aux, Dst, S) =  
  move(0, Aux, Dst,  
    move(s 0, Src, Dst,  
      move(0, Src, Aux, S))) .  
eq Hanoi(s s 0, Src, Aux, Dst, S) =  
  move(0, Src, Dst,  
    move(s 0, Aux, Dst,  
      move(0, Aux, Src,  
        move(s s 0, Src, Dst,  
          move(0, Dst, Aux,  
            move(s 0, Src, Aux,  
              move(0, Src, Dst, S))))))) .
```

Induced Tower of Hanoi Rules (3 examples, 0.076 sec)

```
Hanoi(0, Src, Aux, Dst, S) = move(0, Src, Dst, S)  
Hanoi(s D, Src, Aux, Dst, S) =  
  Hanoi(D, Aux, Src, Dst,  
    move(s D, Src, Dst,  
      Hanoi(D, Src, Dst, Aux, S)))
```

Lernen produktiver Regelmengen mit IGOR2

Derzeitiger Stand

- Beispiele werden per Hand im notwendigen Format angegeben
- Zu tun: Kombination mit einem Problemlösealgorithmus, um Traces zu erzeugen und automatisches Rewriting in das von IGORII benötigte Format (vgl. Schmid, 2003)
- Zu tun: Abruf geeigneter Regelmengen für aktuell gegebene Probleme (z.B. nach Zielen)

Lernen produktiver Regelmengen mit IGOR2

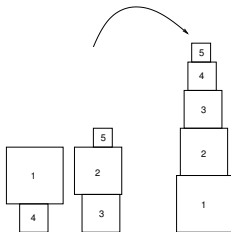
Kognitive Interpretation

- Geht über Lernen als *Chunking* oder als Modifikation von Stärkewerten von Regeln hinaus (typisch in kognitiven Architekturen wie ACT-R, Soar)
- Realisiert Lernen neuer Problemlösestrategien
- Erworbene rekursive Regelmengen entsprechen Problemlöseschemata (ausführbar **und** verbalisierbar, in wissensbasierte Systeme integrierbar)
- IGORII als “Beobachter” des Arbeitsgedächtnisses: wenn der Inhalt generalisierbar ist, wird generalisiert (modelliert “Bewusstwerdung” einer Problemlösungsstrategie, plötzliche Einsicht?)

Turmbau in der Blockswelt

Das *Tower* Beispiel

- Sogar kleine Kinder lernen schnell, wie man Blöcke in einer vorgegebenen Reihenfolge zu einem Turm stapeln kann
- Keine “doofe” Strategie, wie: Lege immer erst alle Blöcke auf den Tisch und baue dann korrekt sortiert auf, sondern optimale Strategie
- IGOR2 lernt *Tower* aus 9 Beispielproblemen von bis zu 4 Blöcken in 1.2 sec



Eines der 9 Beispiele

```
eq Tower(s s table,
  ((s s s s table) (s table) table | ,
  (s s s table) (s s table) table | ,
  nil)) =
put(s s table, s table,
  put(s s s table, table,
    put(s s s s table, table,
      ((s s s s table) (s table) table | ,
      (s s s table) (s s table) table | ,
      nil))))))
```

- Beispiele sind Gleichungen, bei denen im Kopf der Regel der aktuelle Zustand spezifiziert wird und im Körper die optimale Aktionsfolge
- zusätzlich vorgegeben: 10 Beispiele für die Prädikate Clear und IsTower als Hintergrundwissen

Generalisierte produktive Regelmenge für *Tower*

```
Tower(0, S) = S if IsTower(0, S)
Tower(0, S) =
  put(0, Sub1(0, S),
      Clear(0, Clear(Sub1(0, S),
                    Tower(Sub1(0, S), S)))) if not(IsTower(0, S))
Sub1(s(0), S) = 0 .
```

Put the desired block x on the one which has to be below y in a situation where both blocks are clear and the blocks up to the block y are already a tower.

Weitere Beispiele

Problemlösen

- Clearblock (4 examples, 0.036 sec)
- Rocket (3 examples, 0.012 sec)
- Tower of Hanoi (3 examples, 0.076 sec)
- Car Park (4 examples, 0.024 sec)

Rekursive Konzepte

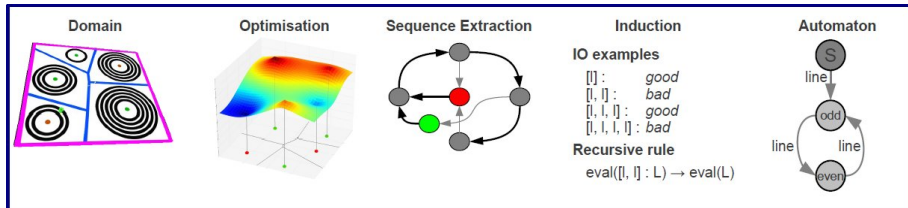
- isa
- odd/even

Grammatikregeln

- Phrase structure grammar

Rekursive Konzepte

- Gerade begonnene Arbeit: Kombination von IGORII mit “embedded cognition” (Roboter-Agent lernt *odd/even* in einer Kreiswelt) (Wernsdorfer, Raab, Schmid & Kitzelmann, AGI 2011)



Lernen einer einfachen Phrasen-Struktur-Grammatik

Lernen von produktiven Regelmengen für die Verarbeitung natürlicher Sprache

- 1: *The dog chased the cat.*
- 2: *The girl thought the dog chased the cat.*
- 3: *The butler said the girl thought the dog chased the cat.*
- 4: *The gardener claimed the butler said the girl thought the dog chased the cat.*

$S \rightarrow NP VP$

$NP \rightarrow d n$

$VP \rightarrow v NP \mid v S$

Zusammenfassung

- Induktives Lernen ist ein zentraler Zugang für den Erwerb neuen Wissens aus Erfahrung
- In höheren kognitiven Bereichen findet Lernen auf der Wissensebene statt
- Menschen können produktive Regelmengen erwerben, die es erlauben, Wissen auf ganze Klassen von Problemen anzuwenden
- Ein möglicher Zugang, um das Lernen produktiver Regelmengen zu modellieren ist analytische induktive Programmierung
- In einem einheitlichen Ansatz können Erwerb von rekursiven Konzepten, sprachlichen Regularitäten sowie Handlungsstrategien modelliert werden
- Diese Art des Lernens geht über das hinaus, was derzeit in kognitiven Architekturen realisiert wird.

Besuchen Sie die KogWis'12 vom 1. bis 3.10. 2012 Bamberg!