

Grundlagen der Kognitiven Informatik

Lernen, Expertise, Analoges Problemlösen

Ute Schmid unterstützt von Michael Siebers

Kognitive Systeme, Angewandte Informatik, Universität Bamberg

letzte Änderung: 30. November 2010

Erwerb von Problemlösefertigkeiten

Learning by Doing

- Macht ein Problemlöser Erfahrung mit einem Problembereich, so kann er dabei Wissen erwerben, das ihm das nachfolgende Lösen ähnlicher Probleme erleichtert
- Lernen aufgrund von Problemlöseerfahrung betrifft vor allem prozedurales Wissen
- Zwei Arten des Lernens:
 - ▶ Wissensoptimierung \leftrightarrow *speed-up effects*
 - ▶ Aufbau neuer Regeln, Strategien, Schemata \leftrightarrow *induktives Lernen*
- *Power Law of Practice* (Newell, 1981): Mit zunehmender Übung werden Probleme aus einem Bereich zunehmend schneller gelöst, was dadurch beschrieben werden kann, dass bei der Suche weniger *Backtracking* notwendig ist, weil die lösungsrelevanten (verstärkten) Regeln direkt angewendet werden.

Verstärkung von Regeln

Tuning in ACT-R

- Mit Produktionsregeln assoziierte Parameter
 - ▶ Stärke
 - ▶ Effekt-Wahrscheinlichkeit
 - ▶ Erfolgswahrscheinlichkeit, Utility
- Die Erhöhung dieser Parameterwerte bewirkt, dass die entsprechende Regel bevorzugt zur Anwendung kommt (Konfliktresolution)
- Maschinelles Lernen: *Reinforcement Learning* – Lernen der Wahrscheinlichkeit mit der eine Aktion in einem bestimmten Zustand ausgeführt wird.

Kombination von Regeln

Production Composition

- Werden zwei Operatoren wiederholt nacheinander angewendet, so können sie zu einem komplexeren Operator zusammengefasst werden.
- In Soar: *Chunking*
- In KI Planung: Macro-Learning (Minton, 1985)
- Bei logisch repräsentierten Operatoren (PDDL) erfolgt die Verknüpfung durch Vereinigung der Anwendungsbedingungen sowie der Vereinigung der ADD- und DEL-Effekte.

Verknüpfung der Bedingungs- und Aktionsteile:

Produktion 1: WENN $\langle B_1 \rangle$ DANN $\langle A_1 \rangle$

Produktion 2: WENN $\langle B_2 \rangle$ DANN $\langle A_2 \rangle$

Verknüpfung: WENN $\langle B_1 \circ B_2 \rangle$ DANN $\langle A_1 \circ A_2 \rangle$.

Utility Problem

- Das Lernen von Makros wurde ca. 1990 – 2008 kaum mehr in Planungssystemen berücksichtigt.
- Ursache: *utility problem* (Minton, 1985)
- Macros reduzieren die Anzahl von notwendigen Interpreter-Zyklen, da ja komplexere Aktionen innerhalb *einer* Regelanwendung ausgeführt werden
- Gleichzeitig erhöht sich aber der Aufwand beim Mustervergleich, da nun neben den gegebenen elementaren Produktionen auch noch eine Menge von Makros auf ihre Anwendbarkeit hin überprüft werden muss.
- Ohne eine vernünftige Heuristik zur Entscheidung, unter welchen Bedingungen Regeln komponiert werden sollen, wird das System leicht mit Makros überschwemmt.
- Neue Ansätze, siehe Planning and Learning Track, ICAPS Planning Competition

Erwerb von Regeln und Strategien

- Die Veränderung von Stärkewerten und Regel-Verknüpfung beschreiben Lernen als Optimierungsprozess.
- Eine „höhere Stufe“ des Lernens ist der Erwerb von neuen Problemlösestrategien.
- Hier wird Wissen über die Lösungsstruktur eines Problembereichs aufgebaut.
- Beispielsweise zeigen Anzai & Simon (1979), dass Probanden beim Lösen eines Turm-von-Hanoi-Problems zunächst die korrekte Folge von Scheibenbewegungen durch heuristische Suche erzeugen, aber bereits nach einigen Durchläufen die rekursive Lösungsstrategie erworben haben.
- Die rekursive Regel beschreibt insbesondere die *Zielstruktur* des Problems.
- Lernen rekursiver Strategien aus Problemlöseerfahrung: Schmid & Wysotzki (AIPS, 2000), Schmid, Hofmann & Kitzelmann (AGI, 2009), Schmid & Kitzelmann (CSR, to appear)

Repräsentation von Strategien

Implizite Repräsentation

- Als Auswahlwahrscheinlichkeit einer Regel gegeben eine Situation
- Modellierung mit Neuronalem Netz oder Reinforcement Learning

Explizite Repräsentation

- Lösungsschema (Novick & Holyoak, 1991)
- Symbolische Regeln (Anderson, 1993)
- Schemata und Regeln sind wechselseitig ineinander überführbar (Rumelhart & Norman, 1981)
- Lernen von Schemata beim analogen Problemlösen: Generalisierung über gemeinsame Struktur (Weller & Schmid, 2008)

Beispiel

Problem: Multipliziere 2 mit 3

Lösung: (* 2 3)

Neues Problem: Addiere 712 mit 91

Mapping: 2/712, 3/91, Multipliziere/Addiere

Neue Lösung: (+ 712 91)

Generalisierung:

Problem: Verknüpfe mit mathematischer Operation Zahl-1 und Zahl-2

Lösung: (math-op zahl-1 zahl-2)

Induktives Lernen

- Generalisierung über beispielhafte Erfahrung entspricht einem *induktiven Schluss* von Beispielen auf eine allgemeine Regel.
- Dabei kann es sich um Regeln zur Zuordnung von Objekten zu semantischen Kategorien handeln oder um Regeln, die beschreiben, auf welche Art ein Problem aus einem bestimmten Wissensbereich gelöst werden kann.
- Induktion wird in der KI vor allem im Bereich Maschinelles Lernen erforscht.
- Zum Erwerb von Regeln (Kontrollwissen) wird beispielsweise die Methode des erklärungsbasierten Lernens verwendet.
- Wie Kontrollwissen zur Lösung von Blockwelt-Problemen aus Beispiellösungen induziert werden kann, beschreiben zum Beispiel Schmid & Kitzelmann (2009)

Problemlösen und Wissen

- In komplexen Problembereichen, wie dem Erstellen von Computerprogrammen, dem mathematischen Beweisen oder dem Steuern komplexer Systeme, ist der Weg vom Novizen zum Experten ein langwieriger Prozess.
- Ein Experte verfügt über reichhaltiges Erfahrungswissen:
 - ▶ Umfassendes Faktenwissen über einen Problembereich, das er effizient organisiert und damit schnell zugreifbar hat,
 - ▶ typische Lösungsmuster für Standardprobleme (Fallbasiertes Schließen),
 - ▶ automatisiertes Handlungswissen (vergleiche prozedurales Wissen),
 - ▶ sowie bereichsspezifische Strategien.
- In vielen Fällen ist er deshalb nicht auf die Suche nach einem Lösungsweg mithilfe schwacher Heuristiken angewiesen.

Intelligente Tutor Systeme

- Zur empirischen Untersuchung des Wissenserwerbs in komplexen Bereichen werden häufig *Intelligente Tutorielle Systeme* (ITS) eingesetzt.
- Damit kann gleichzeitig eine systematische und individuelle Vermittlung von Wissen und Fertigkeiten in einem Bereich erfolgen, und Annahmen über den Erwerb von Problemlösefertigkeiten, die in der tutoriellen Komponente realisiert sind, können überprüft werden.
- Im Kontext der ACT-Theorie wurden einige tutorielle Systeme entwickelt, beispielsweise für den Erwerb von Programmierkenntnissen in Lisp.
- Ein bekanntes Beispiel zur Untersuchung von Expertenwissen sind die Experten-Novizen Vergleiche im Bereich Schach (Chase & Simon, 1973)

Expertensysteme

- In der KI wurde, vor allem in den achtziger Jahren, versucht, Expertenwissen in Expertensystemen umzusetzen.
- Bekannt wurde beispielsweise das System Mycin zur medizinischen Diagnose.
- Schachcomputer wie das System *Deep Blue* sind ebenfalls ein Beispiel für Expertensysteme.
- Problematisch am Aufbau von Expertensystemen ist vor allem die Wissensdiagnose. Häufig werden direkte Verfahren wie Interviews oder Struktur-lege-Techniken verwendet, um Aufschluss über das Wissen eines Experten zu erhalten.
- Dabei ist man darauf angewiesen, dass Experten ihr Wissen verbalisieren können.
- Expertentum zeichnet sich aber gerade durch einen hohen Anteil an prozeduralem und strategischem Wissen aus, das nicht unbedingt dem bewussten Reflektieren zugänglich ist.

Expertensysteme

- Ein alternativer Zugang ist es, Experten mit typischen Problemen zu konfrontieren und zu versuchen, die verwendeten Problemmerkmale und Lösungsstrategien durch indirekte Methoden (maschinelles Lernen) zu ermitteln.
- Beispiel: Experten in der medizinischen Diagnostik werden Röntgenbilder vorgelegt und mithilfe von Klassifikationsverfahren wird ermittelt, auf welche Merkmale die Mediziner ihr Urteil über eine vorliegende Herzerkrankung stützen.
- Expertensysteme basieren im Wesentlichen auf formal repräsentiertem Wissen über einen Bereich und sind in der Lage, mit Hilfe von Deduktionsregeln Schlüsse aus diesem Wissen abzuleiten.

Analoges Problemlösen

- , Bisher wurde dargestellt, wie Problemlösen durch Suche im Problemraum modelliert werden kann.
- Der Problemlöser kann dabei auf Allgemeinwissen zurückgreifen und Heuristiken benutzen, die ihm helfen, die Suche in Richtung des Zielzustands zu steuern.
- Die Suche nach einer Problemlösung kann auf Irrwege führen und ist häufig mit hohem (kognitivem) Aufwand verbunden.
- In vielen Fällen weist ein Problem Ähnlichkeit zu einem bereits bekannten Problem auf.
- Wird dies erkannt, so kann Suche vermieden und stattdessen versucht werden, die Lösung des bekannten Problems auf das neue Problem zu übertragen.
- Man spricht dann von *analogem Problemlösen*.

Die Structure Mapping Engine

- Eines der prominentesten Systeme zur Modellierung des analogen Problemlösens ist die *Structure Mapping Engine* (SME) (Falkenhainer, Forbus & Gentner, 1989)
- Realisierung der Struktur-Vergleichs Theorie von Gentner (1983)
- Fokus: Mapping und Transfer

Teilprozesse

- Representation (Problem: Flexibilität; $on(a,b)$ vs. $below(b,a)$)
- Retrieval (Flaschenhals: Oberflächenanalogien)
- Mapping
- Inference/Transfer
- Learning

Gentners Theorie des Strukturvergleichs

- Wissensbereiche (*domains*) werden als relationale Strukturen repräsentiert.
- Beispiel: Ausschnitt des Wissens über das Sonnensystem dargestellt
- Ein Wissensbereich besteht aus:

Objekten: Als Grundelemente, beispielsweise *Sonne*, *Planet-i*, *Planet-j*.

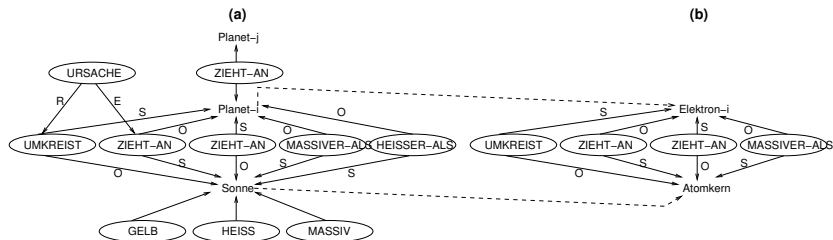
Attributen: Als einstellige Relationen über Objekten, die Objekteigenschaften beschreiben, beispielsweise *gelb(Sonne)*, *heiss(Sonne)*, *massiv(Sonne)*.

Relationen: Zur Beschreibung von Beziehungen zwischen Objekten (Relationen erster Ordnung), beispielsweise *zieht-an(Sonne, Planet-i)*, oder zur Beschreibung von Beziehungen zwischen Relationen (Relationen höherer Ordnung), beispielsweise *ursache(zieht-an(Sonne, Planet-i), umkreist(Planet-i, Sonne))*.

Gentners Theorie des Strukturvergleichs

- Ein Wissensbereich wird also als propositionales Netzwerk repräsentiert.
- In der graphischen Darstellung wird dabei von den Relationen ausgehend mit gerichteten Kanten (Pfeilen) auf deren Argumente verwiesen.
- Die Position eines Arguments in einer Relation wird durch eine Markierung (*label*) angegeben.
- Beispielsweise wird für die Proposition *zieht-an(Sonne, Planet-i)* die Kante zur Sonne mit *S* (Subjekt) und die Kante zum Planeten mit *O* (Objekt) markiert.

Rutherford Analogie



Repräsentation eines Ausschnitts von Wissen über ein Sonnensystem (a) und ein Atom (b) als Struktur aus Objekten und Relationen zusammen mit einer strukturerhaltenden Abbildung für die Rutherford-Analogie “Das Atom ist wie das Sonnensystem” (nach Gentner, 1983).

Mapping

- Die Grundannahme der Theorie des Strukturvergleichs besagt, dass Analogie eine Abbildung (*mapping*) des Wissens eines Bereichs (Basis, *base* oder *source*) auf einen anderen Bereich (Ziel, *target*) ist.
- Dabei wird angenommen, dass die Beziehungen zwischen Objekten in der Basis auch im Zielproblem gelten, während von den Objekteigenschaften abstrahiert wird.
- In der Mathematik werden strukturerhaltende Abbildungen als Homomorphismen bezeichnet.
- Beispielsweise ist dem Rutherfordschen Atommodell die Analogie *Das Wasserstoffatom ist wie unser Sonnensystem aufgebaut* zugrundegelegt.
- Werden das Objekt *Sonne* auf das Objekt *Atomkern* und das Objekt *Planet-i* auf das Objekt *Elektron-i* abgebildet, kann aus dem vorhandenen Wissen über das Sonnensystem etwa inferiert werden, dass das Elektron den Atomkern umkreist wie ein Planet die Sonne.

Mapping

- Zudem kann durch das Übertragen einer Relation zweiter Ordnung inferiert werden, dass das Elektron den Atomkern umkreist, weil es vom Atomkern angezogen wird. Wissen aus einem Bereich kann also benutzt werden, um Beziehungen, die in einem anderen Bereich gelten, zu erklären.
- Nicht übertragen werden dagegen Attribute wie beispielsweise, dass der Atomkern so heiß ist wie die Sonne.

- Im Allgemeinen sind für den Basis-Bereich zahlreiche Beziehungen zwischen den Objekten bekannt, insbesondere auch Relationen höherer Ordnung wie Ursache-Wirkungs-Beziehungen.
- Für den Ziel-Bereich sind dagegen nur die Basisobjekte, eventuell mit einigen Attributen, und einige Relationen (erster Ordnung) bekannt.
- Die Theorie des Strukturvergleichs betont also die *syntaktische Struktur* eines Wissensbereichs.
- Semantische Ähnlichkeiten, also eine Übereinstimmung von Objekten bezüglich ihrer Eigenschaften, werden, anders als etwa im Ansatz von Hummel & Holyoak (1997, LISA), außer acht gelassen.
- Gentner (1983) definiert Randbedingungen für die Abbildung von Relationen auf bereits bekannte Relationen im Ziel und für die Inferenz von Relationen, also deren Übernahme ins Ziel.

Mapping Constraints

Abbildung erster Ordnung: Objekte aus dem Basis-Bereich können auf anders benannte Objekte aus dem Ziel-Bereich abgebildet werden. Aber Relationen aus dem Basis-Bereich müssen auf gleichnamige Relationen aus dem Ziel-Bereich abgebildet werden.

Das heißt, *Sonne* kann auf *Atomkern* abgebildet werden, aber *zieht-an*(x, y) muss auf *zieht-an*(x', y') abgebildet werden.

Isomorphie: Eine Menge von Objekten aus dem Basis-Bereich muss eineindeutig auf die Objekte aus dem Ziel-Bereich abgebildet werden.

Das heißt, *Sonne* kann nur *entweder* auf *Atomkern* oder auf *Elektron-j* abgebildet werden, aber nicht auf beide Objekte und es können nicht *Sonne* und *Planet-i* beide auf *Atomkern* abgebildet werden. Mathematisch spricht man hier von einer bijektiven Abbildung.

Strukturelle Konsistenz: Für eine gegebene Abbildung von Objekten aus dem Basis-Bereich auf Objekte aus dem Ziel-Bereich muss gelten, dass die Abbildung einer Relation verträglich mit den Argumenten erfolgen muss. Mathematisch ist dies die Bedingung für eine homomorphe Abbildung.

Das heißt, wenn *Sonne* auf *Atomkern* abgebildet wurde und *Planet-i* auf *Elektron-i*, dann *muss* die Relation *zieht-an*(*Sonne*, *Planet-i*) auf *zieht-an*(*Atomkern*, *Elektron-i*) abgebildet werden. Eine Abbildung auf *zieht-an*(*Elektron-j*, *Atomkern*) wäre beispielsweise nicht struktur-verträglich.

Systematizität: Es werden bevorzugt möglichst große relationale Gefüge in Kontrast zu einzelnen Relationen abgebildet, insbesondere solche Relationen, die über Relationen höherer Ordnung miteinander verbunden sind (*systematicity principle*).

Die Structure Mapping Engine

Paarweise Zuordnung: Erzeugung aller möglichen paarweisen Zuordnungen zwischen Objekten von Basis zu Ziel und Relationen von Basis zu Ziel sowie einer Liste der Relationen in der Basis, die nicht im Ziel existieren (und die möglicherweise auf das Ziel übertragen werden).

Globale Zuordnung: Kombination der paarweisen Zuordnungen zu maximal konsistenten Strukturen sowie Übernahme von Basis-Relationen ins Ziel (*candidate inferences*).

Bewertung und Auswahl: Für jede globale Zuordnung wird eine Bewertung (*structural evaluation score*) bestimmt und die am besten bewertete Zuordnung wird ausgewählt. Zentraler Bestandteil der Bewertung ist die relative Größe der globalen Zuordnung.

- Formal entspricht dieses Vorgehen der Identifikation einer maximalen Clique in einem Kompatibilitätsgraphen.
- Eine Veranschaulichung der Arbeitsweise des Algorithmus ist für die Rutherford-Analogie dargestellt.
- Die größte globale Zuordnung ergibt sich, wenn *Sonne* auf *Atomkern* und *Planet-i* auf *Planet-j* abgebildet werden.
- Die mit den Relationen *zieht-an(Sonne, Planet-i)* und *umkreist(Planet-i, Sonne)* verbundene Relation *ursache* wird als Inferenz in den Ziel-Bereich übernommen.

Rutherford Analogie in SME

(Elemente der besten globalen Zuordnung sind fett markiert)

- | | | | |
|----|--|---|-------------------------------------|
| 1 | Sonne – Atomkern | 2 | Sonne – Elektron-i |
| 3 | Planet-i – Atomkern | 4 | Planet-i – Elektron-i |
| 5 | Planet-j – Atomkern | 6 | Planet-j – Elektron-i |
| 7 | umkreist(Planet-i, Sonne) –
umkreist(Elektron-i, Atomkern) | | |
| 8 | zieht-an(Sonne, Planet-i) –
zieht-an(Atomkern, Elektron-i) | | |
| 9 | zieht-an(Sonne, Planet-i) –
zieht-an(Elektron-i, Atomkern) | | |
| 10 | zieht-an(Planet-i, Sonne) –
zieht-an(Atomkern, Elektron-i) | | |
| 11 | zieht-an(Planet-i, Sonne) –
zieht-an(Elektron-i, Atomkern) | | |
| 12 | massiver-als(Sonne, Planet-i) –
massiver-als(Atomkern, Elektron-i) | | |
| 13 | heisser-als(Sonne, Planet-i) | | |
| 14 | ursache(zieht-an(Sonne, Planet-i),
umkreist(Planet-i, Sonne)) –
Inferenz: ursache(zieht-an(Atomkern, Elektron-i),
umkreist(Elektron-i, Atomkern)) | | |

Weitere Modelle des Analogen Problemlösens

- Bekannt wurde vor allem ACME (*Analogical Constraint Mapping Engine* (Holyoak & Thagard, 1989) und das Nachfolgesystem LISA (Hummel, 1997).
Bei diesen Systemen wird vor allem die Rolle von semantischen und pragmatischen Constraints beim Mapping betont.
- CASCADE (van Lehn, 1992) ist ein Modell des Problemlösens und Lernens, das auf der Selbsterklärung von Musterlösungen und analogem Transfer von Lösungen basiert.
- Auch im System ACT-R wird versucht, analoges Lernen aus Problemlöseerfahrung zu modellieren. In einer Variante des Systems PUPS (Anderson & Thompson, 1989) wird gezeigt, wie Programmieraufgaben durch analogen Transfer von in Schemata repräsentierten Lösungen gelöst werden können.

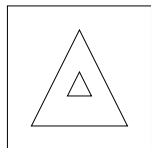
Weitere Modelle des Analogen Problemlösens

KI Systeme

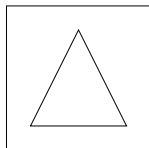
- Das älteste System ist *Analogy* (Evans, 1968), das in der Lage ist geometrische Intelligenztest-Aufgaben zu lösen. Solche Analogien der Form " $a : b$ verhält sich wie " $c : ?$ " werden auch **Proportionalanalogien** genannt.
- Diese Klasse von analogen Problemen werden auch in den neueren KI-Systemen PAN (O'Hara, 1992) und Copycat (Hofstadter, 1995) behandelt.
- Ein KI-System, das Analoges Problemlösen und Lernen kombiniert ist AvA (Analogy via Abstraction), Weller & Schmid, 2008
- In der KI wird anstelle von analogem Problemlösen häufig der Spezialfall des **fallbasierten Schließens** betrachtet, bei dem nur Probleme eines fest vorgegebenen Wissensbereichs miteinander in Beziehung gebracht werden.

Beispielaufgabe für das Evans System

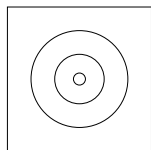
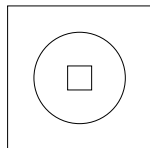
A



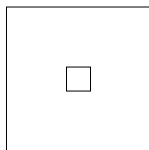
B



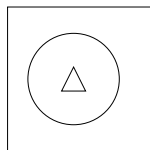
C



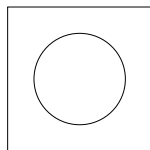
1



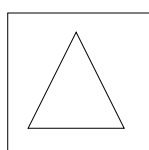
2



3



4



5