UNIVERSITÄT DORTMUND
Fachbereich Informatik

Lehrstuhl VIII
Künstliche Intelligenz

Experimentelle Analyse zweier logik–basierter Lernverfahren

LS–8 Report 6

Guido Lindner und Ursula Robers

Dortmund, 21. Februar 1994

Universität Dortmund
Fachbereich Informatik

University of Dortmund
Computer Science Department

Forschungsberichte des Lehrstuhls VIII (KI),
Fachbereich Informatik
der Universität Dortmund

ISSN 0943-4135

Anforderungen an:

Universität Dortmund
Fachbereich Informatik
Lehrstuhl VIII
D-44221 Dortmund

Research Reports of the unit no. VIII (AI)
Computer Science Department
of the University of Dortmund

ISSN 0943-4135

Requests to:

University of Dortmund
Fachbereich Informatik
Lehrstuhl VIII
D-44221 Dortmund

e-mail: reports@ls8.informatik.uni-dortmund.de
ftp: kimo.informatik.uni-dortmund.de:pub/Reports

Experimentelle Analyse zweier logik–basierter Lernverfahren

LS–8 Report 6

Guido Lindner und Ursula Robers

Dortmund, 21. Februar 1994

Universität Dortmund
Fachbereich Informatik

Zusammenfassung

Ein entscheidendes Problem des logik–basierten Lernens liegt in der Größe des Hypothesenraumes. Möglichkeiten der Einschränkung sind das heuristische Durchsuchen eines vollständigen Hypothesenraums (z.B. FOIL) oder das vollständige Durchsuchen eines eingeschränkten Hypothesenraums (z.B. RDT). Während die theoretische Analyse die Lernbarkeit untersucht, wollen wir durch einen experimentellen Vergleich von RDT und FOIL feststellen, wie sich die unterschiedlichen Einschränkungen des Suchraums in der Praxis auswirken. Für unsere Experimente haben wir zum einen den KRK–Sachbereich und außerdem einen neu modellierten Sachbereich, die Wohnortwahl für Studenten, verwendet.

1 Einleitung

Logik–basiertes Lernen zeichnet sich dadurch aus, daß das Lernergebnis in eingeschränkter Prädikatenlogik vorliegt, von einem System interpretierbar ist und die Verfahren Hintergrundwissen nutzen können. Daraus ergibt sich einerseits eine sehr große Ausdrucksstärke der Lernergebnisse, andererseits wird dadurch der Hypothesenraum sehr groß. Möglichkeiten der Einschränkung sind das heuristische Durchsuchen eines vollständigen Hypothesenraums (z.B. FOIL [Quinlan, 1990]) oder das vollständige Durchsuchen eines eingeschränkten Hypothesenraums (z.B. RDT [Kietz and Wrobel, 1991]). Während die theoretische Analyse die Lernbarkeit untersucht [Haussler, 1990, Kietz, 1992], wollen wir durch einen experimentellen Vergleich von RDT und FOIL feststellen, wie sich die unterschiedlichen Einschränkungen des Suchraums in der Praxis auswirken. Zahlreiche Veröffentlichungen dokumentieren experimentelle Untersuchungen einzelner Lernverfahren anhand von Anwendungen, darunter z.B. RDT [Morik and Sommer, 1993], FOIL [Quinlan, 1990] und GOLEM [Dolšak and Muggleton, 1992]. Für unsere Experimente haben wir zum einen den KRK–Sachbereich (siehe Abschnitt 3.1) ausgewählt, der auch schon von Lavrač und Džeroski in einem Vergleich zwischen FOIL und LINUS benutzt wurde [Lavrač and Džeroski, 1992], und außerdem einen neu modellierten Sachbereich, die Wohnortwahl für Studenten. Bei dem Test–Szenario Wohnortwahl liegt der Schwerpunkt auf der Anwendbarkeit der Verfahren in einem realistischen Sachbereich, während im KRK–Sachbereich die Variation möglicher Einflußfaktoren auf das Lernergebnis im Vordergrund stand.

Im nächsten Abschnitt geben wir einen Überblick über die möglichen Einflußfaktoren auf Lernsysteme und führen die von uns aufgestellten Bewertungskriterien für die Lernergebnisse ein. Im dritten Abschnitt beschreiben wir die Experimente im KRK-Szenario und im vierten Abschnitt die Experimente im Wohnort–Szenario. Im letzten Abschnitt diskutieren wir die Ergebnisse unserer Testreihen und zeigen mögliche Richtungen zur Verbesserung der betrachteten Verfahren auf.[1]

2 Analyse von Lernverfahren

Ziel unserer Analyse ist es, folgende Fragen zu untersuchen:

- Ist die Suche bei einer deklarativen Einschränkung des Hypothesenraums weniger zeitaufwendig, da sie sich jeweils speziell auf eine Anwendung bezieht?
- Wird das korrekte Lernergebnis bei einer heuristischen Einschränkung häufiger verfehlt?
- Welches Verfahren kommt mit einer großen Anzahl von Fakten besser zurecht?
- Beeinflußt die Stelligkeit von Prädikaten den Lernerfolg?
- Inwieweit verbessert vorhandenes Hintergrundwissen das Lernergebnis?
- Welche Art der Einschränkung bewährt sich besser in einer realen Modellierungssituation?

[1] Eine ausführlichere Darstellung findet man in [Projektgruppe-205, 1993].

Zur Klärung dieser Fragen werden die Lernverfahren RDT und FOIL experimentell anhand von zwei Sachbereichen mit unterschiedlicher Variierung der Einflußfaktoren, die in Abschnitt 2.1 beschrieben werden, untersucht. Die Auswertung erfolgt nach den in Abschnitt 2.2 aufgestellten Kriterien.

2.1 Einflußfaktoren

Um zwei Lernverfahren vergleichen zu können, muß man sich zunächst über die Einflußfaktoren auf die Lernergebnisse klar werden. Die fünf wesentlichen Faktoren, die das Lernergebnis jedes Lernverfahrens beeinflussen, sind

- die Beispielmenge,
- die Repräsentation des Wissens,
- der Lernalgorithmus selbst,
- das verwendete Hintergrundwissen und
- das angestrebte Lernergebnis.

2.1.1 Beispielmenge

Die für einen Lernlauf gewählte Beispielmenge kann in verschiedenster Weise variiert werden. Zunächst ist die Frage zu stellen, inwieweit vollständige Beispiele für einen Lernlauf vorliegen, d.h. sind alle Fakten, die zu einem Beispiel gehören, vorhanden. Weiterhin ist die Anzahl der Beispiele wie auch die Verteilung beliebig variierbar. Man kann versuchen, die Beispiele gemäß ihrer in der Realität vorkommenden Verteilung auszuwählen oder mit abweichenden Verteilungen positiver und negativer Beispiele experimentieren. Neben der Verteilung positiver und negativer Beispiele interessiert aber auch, ob für einen gesamten Lernlauf auch wirklich alle in der Realität vorkommenden Fälle berücksichtigt worden sind und wenn, ob deren Verteilung mit der Realität übereinstimmt.

Die Variierung der Beispielanzahl ist der wohl bedeutenste Einflußfaktor für Lernverfahren. Für die Bewertung der Beispielanzahl gibt es ein Maß, die Streuung, die die Anzahl der zur Verfügung stehenden Beispiele für einen Lernlauf ins Verhältnis setzt zu der tatsächlichen Extension des zu lernenden Begriffs.

$$Streuung := 1 - \frac{\mid input(c) \mid}{\mid tat\_ext(c) \mid}$$

wobei $tat\_ext(c)$ die Menge aller Beispiele ist, die dem Begriff c in der Realität angehören und $input(c)$ die Menge der Trainingsbeispiele, die dem System für den Lernvorgang zur Verfügung stehen.

Viele Anwendungen sind dadurch gekennzeichnet, daß nur eine begrenzte Anzahl von Beispielen vorliegt, aus denen Regeln gelernt werden sollen, die später auftretende Situationen korrekt behandeln. In diesem Fall muß berücksichtigt werden, daß die vorhandenen Beispiele sowohl für die Beispielmenge als auch für die Testmenge ausreichen müssen. Ein Verfahren, welches die benötigten Beispiele aus den vorhandenen Beispielen zieht, ist die *cross validation* [Breiman et al., 1984].

2.1.2 Repräsentation des Wissens

Bei einer gewählten Repräsentation für ein Lernverfahren sind in erster Linie folgende Fragen zu stellen:

- Welche Klasse von Beispielsprachen ist zur Darstellung der Beispiele verwendet worden?
- Welchen Hypothesenformalismus verwendet das Lernverfahren?

Mögliche Einschränkungen des Repräsentationsformalismus für Beispiele und/oder Hypothesen sind

- variablenfreie Terme,
- funktionsfreie Terme,
- rekursionsfreie Klauseln,
- generative Klauseln, d.h. jede Variable aus der Konklusion muß auch in der Prämisse vorkommen,
- deterministische Klauseln, d.h. alle Terme der Klausel sind deterministisch verbunden,
- ij–deterministische Klauseln, d.h. deterministische Klauseln, deren maximale deterministische Tiefe i und deren maximale Stelligkeit der Prädikate j ist,
- Reduzierung auf Aussagenlogik,
- Klauseln, deren Form vom Benutzer gegebenen Schemata entsprechen.

Die Repräsentationssprache (also die verwendeten Prädikate einer Anwendung) kann für die Qualität der Lernergebnisse von Bedeutung sein. Es gibt verschiedene Variationsmöglichkeiten für die Repräsentation. Z.B. kann die Stelligkeit der Prädikate verändert werden. Einige Lernverfahren kommen besser mit Prädikaten geringer Stelligkeit zurecht. Andere sind in der Lage, auch Prädikate mit hoher Stelligkeit zu bearbeiten.

2.1.3 Lernalgorithmus

Ein weiterer Punkt ist die Frage nach der Einflußmöglichkeit des Benutzers auf das Akzeptanzkriterium einer Hypothese. Verschiedene Lernalgorithmen bieten hier dem Benutzer über Parameter die Möglichkeit der Einflußnahme auf das Lernergebnis. Eine solche Parametrisierung des Lernalgorithmus macht ihn natürlich nicht leicht handhabbar für den Benutzer. Allerdings wird so meist eine größere Flexibilität in Bezug auf die Lösbarkeit von verschiedenen Lernaufgaben erreicht.

2.1.4 Hintergrundwissen

Vom Hintergrundwissen ist es abhängig, wie leicht ein Begriff zu lernen ist. Durch zusätzliches Hintergrundwissen kann es zu einer Verkürzung der Begriffsbeschreibung kommen. Dies führt dann eventuell dazu, daß neue Begriffe gelernt werden können. Wenn keine Verkürzung erreicht werden kann, ist das Hintergrundwissen eher hinderlich für den Lernvorgang, da zusätzliche Prädikate berücksichtigt werden müssen.

2.1.5 Lernziel

Das Lernziel charakterisiert sich durch die Komplexität der Regelmenge, die zur Beschreibung des zu lernenden Begriffs notwendig ist. Wenn z.B. *eine* Regel ausreicht, also keine verschiedenen Fälle zu unterscheiden sind, so ist das Lernziel einfacher zu erreichen, als wenn eine komplexe Regelmenge zur Beschreibung des Begriffs notwendig ist.

2.2 Bewertungskriterien für Lernergebnisse

In diesem Abschnitt wollen wir einige Bewertungskriterien aufzeigen, die wir in unseren Tests verwendet haben. Zuerst sind da die klassischen Kriterien Vollständigkeit und Korrektheit des Lernergebnisses zu nennen. Hinzu kommen die Zeit, die der Algorithmus zur Berechnung benötigt, und die Redundanzen in der Regelmenge. Diese Kriterien sind eindeutig berechenbar und daher leicht zu bewerten und vergleichen. Anders ist es mit der Sinnhaftigkeit und der Einfachheit der Regeln. Hier sind die Vorstellungen des Benutzers von Bedeutung.

- **Vollständigkeit:** Unter Vollständigkeit verstehen wir die vollständige Ableitung aller tatsächlichen Extensionen eines Begriffs aus dem Sachbereich.

$$\frac{|\ tat\_ext(c) \cap ext(c)\ |}{|\ tat\_ext(c)\ |}$$

  wobei $ext(c)$ die Menge aller Beispiele ist, die durch das Lernergebnis als c klassifiziert werden.

- **Korrektheit:** Werden keine Extensionen im Widerspruch zur tatsächlichen Extension abgeleitet, bezeichnen wir eine Hypothese als korrekt.

$$1 - \frac{|\ not(tat\_ext(c)) \cap ext(c)\ |}{|\ ext(c)\ |}$$

- **Lernzeiten:** Lernzeiten sind schwer zu bewerten. Hierzu muß man sicherstellen, daß die verschiedenen Lernverfahren in der gleichen Testumgebung angewendet werden.

- **Sinnhaftigkeit:** Die gelernten Regeln sollen für den Benutzer sinnhaft sein. Dazu reicht es nicht, daß die Beispielmenge vollständig und korrekt durch die Regeln beschrieben wird. Unserer Meinung nach können Regeln trotzdem von einem Benutzer verworfen werden, wenn sie für ihn nicht einsichtig und nachvollziehbar sind.

- **Einfachheit:** Unter dem Kriterium Einfachheit wollen wir sowohl den Aufbau der gelernten Regeln als auch den Aufbau der gesamten Regelmenge betrachten. Die Regeln sollten möglichst kurz sein. Die Bewertung des Aufbaus der Regelmenge ist von der Zielsetzung im Sachbereich abhängig. Z.B. kann die Zielsetzung darin bestehen, eine möglichst knappe Beschreibung eines Begriffs zu finden.

- **Redundanzen:** Eine Redundanz liegt vor, wenn die Extension einer Regel R eine Teilmenge einer anderen Regel R' ist.

Zusätzlich haben wir zwei weitere Maße für Vollständigkeit und Korrektheit, die die Vorhersagekraft (Streuung) einbeziehen. Wenn z.B. die Testumgebung nicht voraussagekräftig ist, also überhaupt nicht gelernt werden kann, so ist das Lernsystem nicht für Unvollständigkeit und Inkorrektheit verantwortlich zu machen.

- **Dichteunabhängige Korrektheit:** Korrektheit * Streuung
- **Dichteunabhängige Vollständigkeit:** Vollständigkeit * Streuung

Um ein Lernergebnis bewerten zu können, muß man alle Bewertungspunkte zusammen betrachten, da meist eine Aufwertung eines einzelnen Kriteriums eine Abwertung eines anderen zufolge hat.

3 Experimente im KRK–Szenario

Der Sachbereich KRK zeichnet sich dadurch aus, daß er vollständig erfaßt und einfach strukturiert, genauer ij–deterministisch, ist. Aus diesem Grund eignet sich dieser Sachbereich besonders, um viele der Fragestellungen aus Abschnitt 2 näher zu beleuchten.

3.1 Sachbereich

Das KRK–Schachendspiel wurde als Beispiel für das Maschinelle Lernen zuerst von Quinlan beschrieben [Quinlan, 1983]. KRK steht für king–rook versus king, d.h. es spielen der weiße König und der weiße Turm gegen den schwarzen König. Das Lernproblem besteht darin, die illegalen Stellungen der drei Figuren auf dem Schachbrett von den legalen abzugrenzen, d.h. der Begriff `illegal` soll gelernt werden. Es existieren insgesamt 64^3 (=262 144) mögliche Stellungen, von denen (unter der Annahme, daß Weiß am Zug ist) ca. 33% als `illegal` zu klassifizieren sind.
Man kann drei Arten illegaler Stellungen unterscheiden:

1. Ein Feld ist mit mehr als einer Figur belegt.
2. Die beiden Könige sind unmittelbar benachbart.
3. Der schwarze König steht im Schach, d.h. er steht entweder auf derselben Spalte oder derselben Zeile wie der weiße Turm, wobei der weiße König nicht zwischen den beiden steht.

Der Fall, daß der weiße König die Schachstellung verhindert, wurde von den Regeln, die Quinlan mit FOIL lernte, nicht berücksichtigt, woraus sich schlechtere Ergebnisse bzgl. der Vollständigkeit und Korrektheit ergaben [Quinlan, 1990]. In unserem Test–Szenario konnte mit FOIL auch dieser Spezialfall gelernt werden.

3.2 Repräsentationen

Für die Experimente benutzten wir folgende zwei Repräsentationen des KRK–Problems:

1. Attribut–Werte–Repräsentation
   Sie zeichnet sich dadurch aus, daß das gesamte Beispiel, d.h. eine komplette KRK–Stellung, durch ein einziges, siebenstelliges Prädikat dargestellt wird. An der ersten Stelle des Prädikates steht eine Beispielnummer, die restlichen sechs Attribute beschreiben die Positionen der drei Figuren auf dem Spielbrett.

2. First–Order–Repräsentation
   Hierbei wird ein Beispiel durch drei Prädikate für die einzelnen Figuren repräsentiert. Durch die Aufspaltung in drei verschiedene Prädikate geht die Information verloren, daß diese drei Figuren zu *einer* Schachstellung gehören. Aus diesem Grund wurde jedem Prädikat eine eindeutige Beispielnummer hinzugefügt.

   ```
   white_king(r,s,e)
   white_rook(t,u,e)
   black_king(v,w,e)
   illegal(e)
   ```

Um den Einfluß des Hintergrundwissens in unseren Testreihen zu überprüfen, erweiterten wir die Repräsentationen durch unterschiedliche Mengen von Hintergrundwissen:

1. Kein Hintergrundwissen

2. Wenig Hintergrundwissen, d.h. die Prädikate

   ```
   adjacent/2   :(<pos>,<pos>)
   less_than/2  :(<pos>,<pos>)
   not_equal/2  :(<pos>,<pos>)
   ```

   wurden voll instanziiert als Hintergrundwissen zur Verfügung gestellt.

3. Viel Hintergrundwissen, d.h. zusätzlich wurden noch die vollständigen Instanziierungen folgender Prädikate als Hintergrundwissen hinzugefügt:

   ```
   same_row_white_rook_black_king/4   : (<pos>, <pos>, <pos>, <pos>)
   same_row_white_king_white_rook/4   : (<pos>, <pos>, <pos>, <pos>)
   same_row_white_king_black_king/4   : (<pos>, <pos>, <pos>, <pos>)
   same_column_white_rook_black_king/4: (<pos>, <pos>, <pos>, <pos>)
   same_column_white_king_white_rook/4: (<pos>, <pos>, <pos>, <pos>)
   same_column_white_king_black_king/4: (<pos>, <pos>, <pos>, <pos>)
   white_rook_reaches_black_king/1    : (<st>)
   same_field/1                       : (<st>)
   adjacent_kings/1                   : (<st>)
   ```

Die für RDT benötigte Menge an Regelmodellen wurde für jeden Lernlauf angepaßt. Die Gesamtmenge umfaßt daher 41 Metaprädikate. Aus diesem Grund geben wir hier nur exemplarisch zwei Metaprädikate für die First–Order–Repräsentation ohne Hintergrundwissen für den Begriff illegal an:

```
ill1(pos1, pos2, ben, il)          : pos1(x1, y1, st) & pos2(x3, y3, st) &
                                     ben(x1, x3) & ben(y1, y3)
                                     --> il(st).
ill2(pos1, pos2, pos3, neq, lth, il): pos1(x, y1, st) & pos2(x, y2, st) &
                                     pos3(x, y3, st) & lth(y2, y1) &
                                     lth(y3, y1) & neq(y3, y2)
                                     --> il(st).
```

3.3 Durchgeführte Tests

Wie im Abschnitt 2.1 beschrieben, werden die genauen Umstände jedes einzelnen Tests durch Variation der Dimensionen eines „Testraums" bestimmt:

Repräsentation: Entweder First–Order–Repräsentation oder Attribut–Werte Repräsentation.

Zielbegriff für den Lernlauf: `illegal`, `schach_sk`, `wk_verhindert_kein_schach`.

Hintergrundwissen: Entweder keins, wenig oder viel.

Anteil positiver Beispiele: 30%, 50% oder 70%.

Gesamtzahl der Beispiele: 10, 100, 500 oder 1000[2].

Lernsystem: RDT oder FOIL.

Bei einer vollständigen Ausschöpfung des Testraums sind 432 Tests durchzuführen. Da für jeden Test, bestehend aus Beispielgenerierung, Lernlauf und Bewertung der gelernten Regeln zwischen 4 Stunden und 2 Tagen anzusetzen sind, war eine vollständige Durchführung aller Tests für uns nicht möglich.

Wir schränkten deshalb den Testraum auf folgende Testreihen ein:

1. Für die First–Order–Repräsentation mit viel und wenig Hintergrundwissen werden alle Kombinationen der Dimensionen **Zielbegriff für den Lernlauf**, **Anteil positiver Beispiele**, **Gesamtzahl der Beispiele** und **Lernsystem** durchprobiert (144 Tests).

2. Für die Attribut–Werte–Repräsentation mit viel und wenig Hintergrundwissen und dem Lernbegriff `illegal` werden alle Kombinationen der Dimensionen **Anteil positiver Beispiele**, **Gesamtzahl der Beispiele** und **Lernsystem** durchprobiert (48 Tests).

Die sich ergebenden 196 Tests erschienen durchführbar. Leider konnten einige Tests mit FOIL gar nicht durchgeführt werden, da das Programm aus Speichermangel abstürzte, andere Tests wurden stark durch die extremen Lernzeiten von RDT verzögert.

[2] Bei einigen Läufen mußte diese Zahl auf 900 gesenkt werden. Auch mit dieser Anzahl mußten jedoch einige Lernläufe nach mehr als 12–stündiger Lernzeit ohne Ergebnis abgebrochen werden.

3.4 Auszug aus den Testergebnissen

Da wir in diesem Rahmen nicht alle Testergebnisse darstellen können, haben wir stellvertretend die Ergebnisse für die First Order Repräsentation und das Lernziel `illegal` in der Tabelle 1 dargestellt, deren Zeilen die Bewertungskriterien (Vollständigkeit$\hat{=}$V; Korrektheit$\hat{=}$K; Dichteunabhängige Vollständigkeit$\hat{=}$DV; Dichteunabhängige Korrektheit $\hat{=}$DK; Lernzeit (Sekunden)$\hat{=}$Z) und deren Spalten die verschiedenen Mengen an Hintergrundwissen enthalten. Die Einträge sind wieder in einer Tabelle dargestellt, wobei die Spalten den Beispielanzahlen, die Zeilen dem Anteil positiver Beispiele für den jeweiligen Begriff an der Gesamtbeispielanzahl entsprechen. Deren Einträge enthalten die Zahlenwerte des jeweiligen Testergebnisses für FOIL (F) bzw. RDT (R).

Sind irgendwo keine Zahlen eingetragen, so steht dort:

—$^{\emptyset}$	Es wurden keine Regeln gelernt.
—*	Die Regeln wurden nicht bewertet, da die Extension leer war.
—^{F}	FOIL stürzte aus Speichermangel ab.

Z.B. bedeutet der gekennzeichnete Tabelleneintrag in Tabelle 1: Das Lernergebnis von RDT war bei 10 Beispielen und wenig Hintergrundwissen mit einer 50/50 Verteilung von positiven und negativen Beispielen bereits ziemlich vollständig, d.h. ca. 88% der tatsächlichen positiven Begriffe werden von den gelernten Regeln abgedeckt, während die von FOIL gelernten Regeln nur ca. 51% abdecken.

3.5 Diskussion der Testergebnisse

Nachdem nun die Testbedingungen und die Bewertungskriterien beschrieben und die unter Berücksichtigung dieser Kriterien durchgeführten Tests dargestellt wurden, sollen nun die Ergebnisse der Experimente anhand der Kriterien bewertet werden.

3.5.1 Repräsentation

Wie schon in Abschnitt 3.2 angesprochen, sollte durch die Tests mit verschiedenen Repräsentationen ermittelt werden, welches Lernsystem mit welcher Repräsentation am besten arbeiten kann.

RDT hat mit keiner der Repräsentationen Probleme, während FOIL mit der First–Order–Repräsentation deutlich bessere Ergebnisse erzielte als mit der Attribut–Werte–Repräsentation.

3.5.2 Hintergrundwissen

Auch dies war ein Punkt, der von uns im Vorfeld kontrovers diskutiert wurde: würde FOIL angebotenes Hintergrundwissen nutzen oder auf die Fakten zurückgreifen, die nur die Stellung beschreiben?

Aus unseren Testreihen ergab sich, daß FOIL das Hintergrundwissen mindestens ebenso effektiv wie RDT nutzt. Außerdem wurde festgestellt, daß RDT bei der Attribut–Werte–Repräsentation weniger durch zusätzliches Hintergrundwissen behindert wurde als bei der First–Order–Repräsentation. Die Begründung liegt in der Art der Hypothesenraumeinschränkung von RDT, bei der die Stelligkeit der Prädikate eine wichtige Rolle spielt.

		kein Hinterg. 500	kein Hinterg. 2000	wenig Hintergrundwissen 10	wenig Hintergrundwissen 100	wenig Hintergrundwissen 500	viel Hintergrundwissen 10	viel Hintergrundwissen 100	viel Hintergrundwissen 500
V	0.3	R0.717 F0.763	R0.717 F—^{F}	R—$^{\emptyset}$ F—*	R0.959 F0.97	R0.98 F0.97	R—$^{\emptyset}$ F0.958	R0.966 F1.0	R1.0 F0.930
	0.5			R0.883 F0.513	R0.987 F0.938	R0.989 F0.996	R0.664 F0.958	R1.0 F1.0	R1.0 F0.949
	0.7			R0.918 F0.562	R0.967 F1.0	R0.987 F0.995	R0.886 F0.958	R1.0 F1.0	R1.0 F1.0
K	0.3	R1.0 F0.984	R1.0 F—^{F}	R—$^{\emptyset}$ F—*	R0.83 F0.987	R1.0 F0.987	R—$^{\emptyset}$ F1.0	R0.987 F1.0	R0.987 F0.988
	0.5			R0.388 F0.494	R0.997 F0.964	R0.996 F0.990	R1.0 F1.0	R0.987 F1.0	R0.987 F0.993
	0.7			R0.346 F0.322	R0.814 F0.989	R0.998 F0.988	R0.412 F1.0	R1.0 F1.0	R0.987 F0.987
DV	0.3	R0.713 F0.758	R0.7 F—^{F}	R—$^{\emptyset}$ F—*	R0.958 F0.969	R0.974 F0.965	R—$^{\emptyset}$ F0.958	R0.965 F0.999	R0.994 F0.925
	0.5			R0.883 F0.513	R0.986 F0.937	R0.983 F	R0.664 F0.958	R0.999 F0.999	R0.994 F0.944
	0.7			R0.917 F0.562	R0.964 F0.999	R0.982 F0.990	R0.885 F0.958	R0.999 F0.999	R0.994 F0.994
DK	0.3	R0.994 F0.979	R0.977 F—^{F}	R—$^{\emptyset}$ F—*	R0.829 F0.986	R0.994 F0.981	R—$^{\emptyset}$ F0.9999	R0.986 F0.999	R0.982 F0.982
	0.5			R0.388 F0.494	R0.995 F0.963	R0.990 F0.985	R0.9999 F0.9999	R0.986 F0.999	R0.982 F0.988
	0.7			R0.346 F0.322	R0.813 F0.988	R0.992 F0.982	R0.412 F0.9999	R0.999 F0.999	R0.982 F0.982
Z	0.3	R5296 F616	R57506 F—^{F}	R6236 F0.3	R5696 F4	R13213 F4	R57 F0.2	R974 F5.7	R7004 F174
	0.5			R3994 F0.1	R5700 F4.7	R12887 F75	R849 F0.1	R974 F5.3	R6990 F178
	0.7			R3874 F0.1	R5763 F4	R12755 F55	R830 F0.2	R1004 F6.4	R7118 F154

Tabelle 1: Tests in der First–Order–Repräsentation über das Lernziel illegal

Interessant ist, daß beide Lernsysteme auch gänzlich ohne Hintergrundwissen noch eine hohe Vollständigkeit bzw. Korrektheit erreichen. Daraus läßt sich ablesen, daß die einfachsten Regeln den größten Teil der tatsächlichen Extension abdecken.

3.5.3 Verteilung der Trainingsbeispiele

Die Verwendung unterschiedlicher Verteilungen hat keine eindeutigen Ergebnisse gebracht. Man kann lediglich sagen, daß beide Lernsysteme ihre Hypothesen in erster Linie aufgrund der positiven Beispiele bilden und daher bei einer extrem kleinen Trainingsmenge ein hoher Anteil an positiven Beispielen günstig ist.

3.5.4 Lernzeit, Speicherbedarf, Betriebssicherheit

Was diese Aspekte angeht, bietet sich ein zwiespältiges Bild. Einerseits liegen die Lernzeiten von RDT um einen derartig extremen Faktor über denen von FOIL (man beachte: 57000 Sekunden sind knapp 16 Stunden), daß man in diesem Bereich von einer deutlichen Überlegenheit von FOIL sprechen kann.

Andererseits erwies sich RDT als das deutlich sicherere System. Wenn die Fakten überhaupt in die Wissensbasis paßten (und das war ja eine Grundvoraussetzung für den Lernlauf bei unserer Implementierung), dann lernte RDT auch.

FOIL dagegen hatte Probleme mit großen Faktenmengen. Ab ca. 5000 Fakten wurde soviel Speicher reserviert ($\geq$ 12 MB), daß auf den uns zur Verfügung stehenden Maschinen (ausgerüstet mit 24 MB) ein Absturz erfolgte. Obwohl man bedenken muß, daß MOBAL[3] mit der gesamten Wissensbasis im Speicher blieb, da der FOIL–Aufruf ja aus MOBAL erfolgte, kann dies nicht als Erklärung dienen: Eine Beobachtung der Speicherauslastung ergab, daß FOIL immer mehr Speicher reservierte, während MOBAL vom System automatisch ausgelagert wurde. Es ergab sich, daß der Speicherverbrauch von FOIL exponentiell (vgl. [Pazzani and Kibler, 1992]), der von RDT aber nur linear mit der Beispielanzahl gewachsen ist.

3.5.5 Redundanz bzw. Regelaufbau

Auch hier fällt es schwer, eine eindeutige Aussage zu treffen. Einerseits lernte RDT große Mengen von Regeln, zum Teil bis zu 40 Regeln für einen Begriff, die kaum zu einer Erhöhung der Vollständigkeit beitrugen, weil sie z.B. in der Prämisse immer wieder die Positionsprädikate permutierten oder sich in einer Repräsentation mit viel Hintergrundwissen auf die Stellungsprädikate bezogen (ein Verhalten, das ja gerade von FOIL erwartet wurde).

Solche redundanten Hypothesen wurden bei FOIL schon wegen der Regelung, daß nur eine bestimmte Maximalzahl an Regeln gelernt werden kann, verhindert. Andererseits lernte FOIL häufig Regeln, deren Prämissen sich auf mehr als eine Stellung bezogen, also keine Aussagekraft haben (vgl. Abschnitt 4.4, Sinnhaftigkeit der Regeln für den Benutzer).

[3]RDT ist Bestandteil des Modellierungssystems MOBAL, das eine Entwicklung der GMD ist.

4 Experimente im Wohnortwahl–Szenario

Wie auch im KRK–Szenario handelt es sich hier um einen strukturierten Sachbereich, der allerdings nicht vollständig erfaßt ist, aber einen realistischeren Hintergrund hat als der KRK–Sachbereich. Insbesondere wollen wir mit diesem Sachbereich versuchen, die Frage nach der Eignung der Verfahren in einer realistischen Modellierungsumgebung zu beantworten, aber auch andere Fragen aus dem Abschnitt 2 sollen in diesem Test–Szenario berücksichtigt werden.

4.1 Sachbereich

Der Sachbereich wurde im Rahmen der Projektgruppe 205 an der Universität Dortmund modelliert. Es geht in diesem Sachbereich um die Wohnortwahl der Studierenden an der Universität Dortmund. Durch eine Befragung von 54 Studierenden wurde nach den Gründen gesucht, die einen Studenten dazu bewegt haben, seinen momentanen Wohnort auszuwählen. Mit Hilfe dieser Datenbasis sollten Regeln gelernt werden, die es erlauben, wohnungssuchenden Studierenden, die Anforderungen an einen zukünftigen Wohnort geäußert haben, Vorschläge zu unterbreiten, welcher Wohnort diesen Anforderungen genügt. Die Signifikanz von gleichzeitigem Auftreten eines erfüllten Wunsches und der Zufriedenheit einer Person soll sich also in den gelernten Regeln zeigen.

Der Sachbereich stellt damit andere Anforderungen an ein Lernverfahren als KRK, da eine vollständige Beschreibung des Problems noch nicht bekannt ist und möglicherweise nicht existiert.

4.2 Repräsentation

Im folgenden soll nun die Repräsentation des Wohnortsachbereichs für die Lernverfahren genauer beschrieben werden.

Für jede der 54 Testpersonen wurden Fakten der Form

freizeit(p,o)	~	Person p schreibt dem Ort o gute Freizeitmöglichkeiten zu
infrastruktur(p,o)	~	Person p schreibt dem Ort o eine gute Infrastruktur zu
...		...

oder ggf. deren Negation eingegeben.

Diese Fakten wurden mit MOBAL statistisch ausgewertet, so daß einer Stadt eine gewisse Eigenschaft generell zugeordnet wurde, wenn mehr als die Hälfte aller Befragten ihr diese zugesprochen haben. Also

hat_eigenschaft(freizeit,o)
hat_eigenschaft(infrastruktur,o)
...

Das Prädikat `wunsch` soll nun ausdrücken, welche Eigenschaften der befragten Person besonders wichtig waren und welche eher nicht. Die Fakten haben die Form:

wunsch(p,freizeit)
wunsch(p,infrastruktur)
wunsch(p,freunde)
wunsch(p,job)
...

Da die sozialen Kontakte nicht als eine Eigenschaft des Ortes angesehen werden können, müssen diese natürlich auch beim Lernen anders berücksichtigt werden als die Eigenschaften der Städte. Dazu wurden Fakten der Form

hat_private_eigenschaft(p,o,freunde)
hat_private_eigenschaft(p,o,job)
...

eingegeben, um auszudrücken, an welchem Ort welche sozialen Kontakte einer Person bestehen.

Das Zielprädikat `zufrieden` gibt den Ort an, an dem eine Person zufrieden ist

zufrieden(p,o).

4.3 Test

Aufgrund der relativ geringen Anzahl der Befragten wurden Test- und Trainigsmengen durch *cross validation* gebildet, ohne dabei die Verteilung positiver und negativer Beispiele zu berücksichtigen. Dabei dienten jeweils 11 bzw. 10 Beispiele als Testmenge, der Rest (43 bzw. 44) als Trainingsmenge. Die fünf Testläufe wurden dann sowohl mit FOIL als auch mit RDT durchgeführt.

4.3.1 Testläufe mit RDT und FOIL

Um die Grundlagen für RDTs Lernvorgang zu schaffen, haben wir drei Metaprädikate vorgegeben, die erlauben, jeweils eine bestimmte Anzahl von Wunsch- und Eigenschaftskombinationen zu lernen.
Beispiel für ein Metaprädikat von RDT:

```
metapraedikat_1(p1,p2,p3,p4,p5,p6,p7,e1,e2,e3,e4,q):
p1(p,e1) & p2(e1,o) & p3(p,e2) &  p4(e2,o) & p5(p,e3) & p6(p,o,e3)
--> q(p,o)
```

Eine Regel sollte akzeptiert weden, wenn mehr als drei Beispiele die Regel bestätigen und weniger als 4 der Regel widersprechen. Zusätzlich sollten weniger als 10% der gesamten Beispiele für die Regel negative Beispiele sein.

Mit diesen Einstellungen von RDT ergaben die Lernläufe die in Tabelle 2 dargestellten Ergebnisse.

Zur Durchführung der FOIL–Tests wurde die Möglichkeit genutzt, FOIL von MOBAL aus zu starten, wobei auf die gleiche Faktenbasis, die auch bei den RDT–Tests benutzt wurde, zurückgegriffen werden konnte. Die Ergebnisse der Lernläufe sind in Tabelle 2 dargestellt.

4.3.2 Interpretation der Ergebnisse bzgl. der Korrektheit:

Durch die Tatsache, daß der Wohnort–Sachbereich kaum negative Beispiele für `zufrieden`–Fakten liefert, ist die Möglichkeit, daß ein inkorrektes `zufrieden`–Fakt abgeleitet wird, fast nicht vorhanden. Es ist also kaum überraschend, daß die Tests ein Korrektheitsmaß von fast 100% bestätigen. Aus diesem Grund berechneten wir die Korrektheit zusätzlich

		Vollst.	Korrekth.	Streu.	dichteunabh. Korrekth.	dichteunabh. Vollst.	cwa–Korrekth.
Test 1	RDT	0.5	1	0.2	0.2	0.1	0.56
	FOIL	0.8	1	0.2	0.2	0.16	0.53
Test 2	RDT	0.56	1	0.2	0.2	0.112	0.71
	FOIL	1	0.99	0.2	0.2	0.2	0.15
Test 3	RDT	0.82	1	0.2	0.2	0.164	1
	FOIL	0.91	1	0.2	0.2	0.182	0.1
Test 4	RDT	0.73	1	0.2	0.2	0.146	0.62
	FOIL	0.82	1	0.2	0.2	0.164	0.5
Test 5	RDT	0.57	1	0.19	0.19	0.1082	0.67
	FOIL	0.57	0.98	0.19	0.19	0.1083	0.13
Mittel–	RDT	0.65	1	0.2	0.2	0.13	0.71
werte	FOIL	0.82	0.99	0.2	0.2	0.164	0.28

Tabelle 2: Testergebnisse

auch unter der closed–world–assumption (cwa–Korrektheit). Alle nicht von den Testpersonen angegebenen Wohnorte wurden für die jeweilige Testperson als `not(zufrieden(p,o))` eingetragen. Dadurch verschlechterte sich das Korrektheitsmaß für FOIL erheblich (siehe Tabelle 2).

4.4 Diskussion der Testergebnisse

Anhand der in Abschnitt 2.2 aufgestellten Bewertungspunkte werden die Ergebnisse der Experimente kurz zusammengefaßt.

Einfachheit der Regelmenge Während FOIL in drei Tests jeweils eine und in den anderen beiden Tests zwei Regeln lernte, bestanden die von RDT erzeugten Regelmengen durchweg aus mehr als 35 Regeln. Die Extensionen der gesamten Regelmengen waren trotzdem wesentlich kleiner als die Extensionen der FOIL–Regeln, die z.T. aus mehr als 100 Fakten bestanden. RDT leitete nur duchschnittlich 9 Fakten ab.

Redundanz In diesem Test–Szenario wurden von beiden Verfahren keine redundanten Regeln gelernt.

Lernzeiten Hier kann im Prinzip die gleiche Aussage gemacht werden wie im KRK–Sachbereich, nämlich, daß die Lernzeiten von RDT wesentlich über denen von FOIL liegen. FOIL brauchte durchschnittlich 94,34 Sekunden für einen Lernlauf, während RDT dafür schon durchschnittlich ca. 3700 Sekunden benötigte.

Sinnhaftigkeit der Regeln für den Benutzer Die Diskrepanz zwischen Korrektheit und cwa–Korrektheit bei FOIL legt nahe, daß die FOIL–Regeln nicht aussagekräftig sind. Und in der Tat bestätigt sich bei genauerem Hinsehen, daß die Regeln, wenn man ihre intendierte Bedeutung berücksichtigt, für einen Benutzer nicht viel Sinn machen.

```
zufrieden(A,B) :- *hat_private_eigenschaft(C,B,D), *wunsch(A,E),
                  ~*hat_private_eigenschaft(C,B,E)
```

Es ist zu bezweifeln, daß eine Person A an einem Ort B zufrieden ist, wenn Person C an dem Ort B beispielsweise einen Job aber keine Familie hat und sich die Person A wünscht, in der Nähe ihrer Familie zu wohnen.

Erklärungsansatz für die unterschiedlichen Ergebnisse bzgl. der Sinnhaftigkeit der gelernten Regeln Es muß beachtet werden, daß der Benutzer von RDT die Möglichkeit erhält, seine Erkenntnisse über die semantischen Strukturen und Zusammenhänge im Sachbereich schon in deren Modellierung einfließen zu lassen, indem er geeignete Metaprädikate vorgibt. In unserem Fall ist es offensichtlich, daß eine direkte Abhängigkeit zwischen einem Wunsch nach einer bestimmten Eigenschaft, deren Ausprägung an einem Ort und der Zufriedenheit der Person mit dem Wohnort besteht. Diese Struktur gaben wir in Form von Metaprädikaten ein, so daß sichergestellt war, wenn RDT eine Regel lernt, entspricht sie der Intention des Benutzers. FOIL dagegen wird in diesem Sachbereich erst brauchbare Regeln lernen, wenn auf eine Datenbasis zurückgegriffen werden kann, aus der die Zusammenhänge deutlicher bzw. eindeutig hervorgehen, d.h. wenn die Qualität der Beispiele erhöht wird.

5 Schlußfolgerungen

Insgesamt muß man eigentlich sagen, daß beide Lernsysteme weit mehr gemeinsam haben, als bei Betrachtung der Algorithmen zu vermuten wäre. Beide Systeme können als so ausgereift betrachtet werden, daß die Qualität ihres Lernergebnisses im wesentlichen vom tatsächlichen Informationsgehalt der Beispielmenge abhängt.

Der größte Unterschied beider Verfahren besteht offensichtlich in den Lernzeiten. Die Abweichung der Lernzeiten von RDT gegenüber denen von FOIL könnte durch die unterschiedlichen Implementierungen (PROLOG – C) begründet werden. Diese Erklärung wäre aber nur ausreichend, wenn es sich um einen konstanten Faktor handeln würde. Aus unseren Testreihen ergibt sich aber ein nicht linearer Unterschied der Lernzeiten. Die Ironie des Schicksals liegt hier darin, daß RDT als Bestandteil eines Systems, das nach dem *sloppy modeling*–Paradigma (siehe [Morik, 1989]) arbeitet, und dank der extrem vielen Möglichkeiten, das Lernverhalten durch Parameter und Metaprädikate zu beeinflussen, sehr oft aufgerufen werden müßte, um das Lernergebnis zu optimieren, was aber durch die extremen Lernzeiten kaum realisierbar ist. FOIL dagegen ist in der Lage fast augenblicklich ein Lernergebnis zu erzielen. Aber es kann kaum durch Parameter o.ä. beeinflußt werden, weshalb eigentlich nur ein einziger Lernlauf sinnvoll ist.

In Bezug auf die Korrektheit und die Verwendung von Hintergrundwissen konnten in unseren Tests zwischen den verschiedenen Einschränkungsarten keine wesentlichen Unterschiede gefunden werden. Beide Verfahren zeichneten sich durch eine sehr hohe Korrektheit aus, wobei das angebotene Hintergrundwissen nahezu gleich gut verwendet wurde.

Eine große Anzahl an Fakten stellt dagegen für beide Verfahren gleichermaßen ein Problem dar. Bei RDT erhöhen sich durch eine große Anzahl von Fakten die Lernzeiten sehr stark, während sich bei FOIL Speicherplatzprobleme ergeben (siehe Abschnitt 3.5.4).

Zum Punkt der Einsatzmöglichkeiten der Verfahren in realen Modellierungssituationen ist zu sagen, daß hier der Wissensingenieur zusätzliches Wissen über den Sachbereich und die zu erwartenden Regeln besitzt. Bei RDT besteht die Möglichkeit, dieses Wissen in Form von Metaprädikaten für das Lernverfahren nutzbar zu machen. Die Vorteile dieser Möglichkeit werden durch die Erfahrungen mit dem Wohnort–Sachbereich bestätigt.

Abschließend ist anzumerken, daß der RDT–Algorithmus automatisch die Bildung bestimmter Sorten von Hypothesen verhindert, die *mit Sicherheit* keinen Informationsgewinn versprechen, z.B. weil es sich um Spezialisierungen bereits akzeptierter Hypothesen handelt. Weiterhin werden nur solche Hypothesen zugelassen, die zu den vom Benutzer eingegebenen Metaprädikaten passen. Alle verbleibenden Hypothesen werden aber einzeln durchprobiert und entweder akzeptiert oder verworfen.

FOIL dagegen verhindert auch die Bildung von Hypothesen, die *wahrscheinlich* keinen Informationsgewinn bringen, weil sie nicht zu einer bestimmten, festgelegten Heuristik passen.

In den Diskussionen, die zur Festlegung der Testreihen führten, gab es nun die Vorstellung, die in FOIL festgelegte Heuristik sei speziell auf die von Quinlan in seinen Artikeln genannten Beispiele zugeschnitten und werde möglicherweise schon bei leichten Änderungen des Szenarios versagen. Diese Ansicht hat sich nicht bestätigt. Wie an den geringen Lernzeiten zu erkennen ist, hat die Heuristik tatsächlich den Hypothesenraum erheblich eingeschränkt, die Bewertungskriterien und eine Betrachtung der gelernten Regeln zeigen, daß bei der heuristischen Beurteilung kaum Fehler aufgetreten sind.

Die Metaprädikate, die RDT verwendet, haben den Sinn, es dem Wissensingenieur zu ermöglichen, sein Expertenwissen in die Hypothesenbildung einzubringen. Im KRK–Sachbereich ergab sich, daß die Hypothesenbildung per Heuristik durchaus ausreichte, während die per Metaprädikat gebildeten Hypothesen noch viel zu viele Redundanzen enthielten. Im Wohnort–Sachbereich reichte die Heuristik allein dagegen nicht aus, um für den Benutzer sinnvolle Regeln zu lernen, da die Anzahl der Beispiele wahrscheinlich für FOIL zu gering war, um eine Beziehung zwischen einem Wunsch und der entsprechenden Eigenschaft zu erkennen. Insgesamt kann man sagen, daß die heuristische Hypothesenbildung durchaus wertvoll für ein Lernverfahren sein kann, und man könnte darüber nachdenken, es RDT zu erlauben, bei der Bildung bzw. bei der Eliminierung von Hypothesen mittels (für den Benutzer sichtbaren) Heuristiken vorzugehen.

6 Danksagung

Die vorgestellten Ergebnisse stammen aus der Arbeit der Projektgruppe 205 „Systematische Analyse von Lernsystemen“ der Universität Dortmund unter der Leitung von Katharina Morik und Anke Rieger. Weiterhin waren U. Aldenhoff, B. Eller, C. Hölscher, S. Lange, S. Lehmke, C. Maiworm, M. Mühlenbrock, T. Nickel, V. Saueressig, R. Schiprowski, F. Timmerbeil und R. Wegner an diesem Projekt beteiligt. Ihnen sei an dieser Stelle nochmals gedankt.

Literatur

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and regression trees.* Wadsworth and Brooks, Pacific Grove.

[Dolšak and Muggleton, 1992] Dolšak, B. and Muggleton, S. (1992). The application of inductive logic programming to finite-element mesh design. In Muggleton, S., editor, *Inductive Logic Programming*, chapter 23, pages 453–472. Academic Press, San Diego, CA 92101, 1st edition.

[Haussler, 1990] Haussler, D. (1990). Probably approximately correct learning. In *Proceedings Eight National Conference on Artificial Intelligence*, pages 1101–1108.

[Kietz, 1992] Kietz, J.-U. (1992). A comparative study of structural most specific generalizations used in machine learning. In *Proceedings ILP Workshop at ECAI.*

[Kietz and Wrobel, 1991] Kietz, J.-U. and Wrobel, S. (1991). Controlling the complexity of learning in logic through syntactic and task-oriented models. In Muggleton, S., editor, *Inductive Logic Programming*, chapter 16, pages 335 – 360. Academic Press, London. Also available as Arbeitspapiere der GMD No. 503, 1991.

[Lavrač and Džeroski, 1992] Lavrač, N. and Džeroski, S. (1992). Inductive learning of relations fron noisy data. In Muggleton, S., editor, *Inductive Logic Programming*, chapter 25, pages 495–516. Academic Press, San Diego, CA 92101.

[Morik, 1989] Morik, K. (1989). Sloppy modeling. In Morik, K., editor, *Knowledge Representation and Organization in Machine Learning*, pages 107–134. Springer Verlag, Berlin, New York.

[Morik and Sommer, 1993] Morik, K. and Sommer, E. (1993). What online Machine Learning can do for Knowledge Acquisition – A Case Study. Arbeitspapiere der GMD, GMD, Schloß Birlinghoven, D–5205 St. Augustin 1, Germany.

[Pazzani and Kibler, 1992] Pazzani, M. and Kibler, D. (1992). The Utility of Knowledge in Inductive Learning. *Machine Learning*, 9(1):57–64.

[Projektgruppe-205, 1993] Projektgruppe-205 (1993). Systematische Analyse eines Lernsystems. Erschienen als interner Bericht des Fachbereichs Informatik, Universität Dortmund.

[Quinlan, 1983] Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. In Michalski, R., Carbonell, J., and Mitchell, T., editors, *Machine Learning - An Artificial Intelligence Approach*, pages 463 – 482. Tioga, Palo Alto, CA.

[Quinlan, 1990] Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning*, 5(3):239 – 266.