Regellernen

Gegeben:

Hypothesensprache für Mengen von Regeln LH

Beobachtungen E in einer Sprache LE

Hintergrundwissen Tin einer Sprache LT

T, E |-/– (Konsistenz)

Ziel:

C ∈ LH, so daß $\mathcal{M}+$ (T, E) ⊆ $\mathcal{M}$ (C) (Gültigkeit)

∀c ∈ C, ∃ e ∈ E: T, E-{e} |-/- e,

aber T, E-{e}, c |-- e (Notwendigkeit)

Wenn c gültig und notwendig ist, dann C |-- c (Vollständigkeit)

Es gibt keine echte Teilmenge von C,

die gültig und vollständig ist (Minimalität)

(minimales) Modell

Gegeben eine Interpretation I für eine Menge von Formeln F.

I ist ein Modell von F, $\mathcal{M}$ (F), wenn alle Formeln von F in I wahr sind.

Wenn es keine Interpretation I' gibt, mit I' ⊆ I und I' ist ein Modell von F, ist I ein minimales Modell für F, $\mathcal{M}+$.

Begriffslernen aus Beispielen

- gegeben: Hypothesensprache LH für den Begriff
- Hintergrundwissen B in einer Sprache LB
- Menge E = P ∪ N in einer Sprache LE
- wobei B, E |≠ , B |≠ E
- Ziel:
- H ∈ LH mit
- B, H, E |-/- (Konsistenz)
- B, H |-- P (Vollständigkeit)
- ∀e ∈ N: B, H |-/- e (Korrektheit)
- Präferenzkriterium:
- z.B.: speziellste Generalisierung, generellste Diskriminierung
- u.a.: triviale Lösung (Aufzählung von P) ausschließen!

Zuverlässige Lernoperatoren

LH wird durch eine Generalisierungsbeziehung geordnet.

Diese soll eindeutig sein:

für zwei Klauseln existiert genau eine Generalisierung (Spezialisierung).

Dann ist Lernen die Suche entlang der Generalisierungsbeziehung im Hypothesenraum.

Der Hypothesenraum kann sicher beschnitten werden:

- Wenn beim Generalisieren bereits die Hypothese C1 ein negatives Beispiel abdeckt, so wird dies auch jede Generalisierung von C1 tun. Von C1 ausgehend braucht nicht generalisiert zu werden.
- Wenn beim Spezialisieren bereits die Hypothese C1 ein positives Beispiel nicht abdeckt, so wird dies auch jede Spezialisierung nicht tun. Von C1 ausgehend braucht nicht spezialisiert zu werden.

Zuverlässige Lernoperatoren liefern die speziellste Generalisierung oder generellste Spezialisierung.

Heuristische Verfahren liefern irgendeine konsistente Hypothese.

Generalisierung: Implikation

Eine Hornklausel C1 ist genereller als eine andere, C2, gdw.

C1 --> C2

C1 ist genereller als C2 bezüglich einer Theorie T , gdw.

T, C1 --> C2

Äquivalenz:

Sei T eine Konjunktion von Hornklauseln, dann sind die Klauseln C1 und C2 logisch äquivalent bzgl. T gdw.

T, C1 --> C2 und T, C2 --> C1

Redundanz :

Ein Literal L ist redundant in einer Klausel C bzgl. T gdw.

C und C \ {L} sind äquivalent bzgl. T.

Beispiel Generalisierung

T: append ([ ], C, C).

C2: append ([1,2], [3], [1,2,3]).

C1: append([A | B], C, [A | E]) :- append (B, C, E).

C1, T --> C2

¬ append ([1,2], [3], [1,2,3]).

append ([A | B], C, [A | E]) , ¬ append (B, C, E).

{A/1, B/[2], C/[3], E/[2,3]}

¬ append ([2 | []], [3], [2,3]).

append ([A | B], C, [A | E]) , ¬ append (B, C, E).

{A/[2], B/ [], C/ [3], E / [3]}

¬ append ([ ], [3], [3]).

append ([ ], C, C).

{C / [3]}

Beweis von C2 mithilfe von T und C1

Generalisierung: Subsumtion

Eine Hornklausel C1 ist genereller als eine andere, C2, gdw.

C1 subsumiert C2.

Ein Literal L1 subsumiert ein Literal L2 gdw.

L1 σ = L2.

Eine Klausel C1 subsumiert eine andere, C2, gdw.

C1$\theta \subseteq$ C2

Eine Hornklausel subsumiert eine andere, gdw.

C1kopf$\theta \subseteq$ C2kopf und

C1körper θ = C2körper

Die Subsumtion ist eine korrekte, aber nicht vollständige Ableitung, d.h.

C1 subsumiert C2 ==> C1 impliziert C2, aber nicht umgekehrt.

Beispiel Generalisierung

C1: member (X, [Y| Z]):- member(X,Z).

C2: member (3, [1,2,3]):- member(3, [2,3]), member(3, [3]).

θ: { X/3, Y/1, Z/ [2,3] }

C1kopf θ = member (3, [1 | [2,3]) = C2kopf

C1körper θ = member (3, [2, 3]) $\subseteq$ C2körper

Nicht alles, was unter der Implikation eine Generalisierung ist, ist es auch unter der Subsumtion!

T: append ([ ], C, C).

C2: append ([1,2], [3], [1,2,3]).

C1: append([A | B], C, [A | E]) :- append (B, C, E).

kein Hintergrundwissen!

Generalisierung kann nicht länger sein als Generalisiertes!

Least General Generalization Plotkin

LGG (C1, C2):

Für alle Paare von Literalen $L1_i \in C1$, $L2_i \in C2$, suche die mit gleichem Prädikatensymbol und gleichem Vorzeichen heraus --

bilde LGG ($L1_i$, $L2_i$)

Die Generalisierung von C1 und C2 ist die Vereinigung aller generalisierten Literale.

Aus dieser Generalisierung werden alle redundanten Literale entfernt.

Generalisierung von Literalen Anti-Unifikation

zwei Literale mit demselben Prädikatsymbol und Vorzeichen

p(s1, ..., sn)   p (t1, ..., tn) von links nach rechts durchgehen

LGG (si, ti) = X, falls si, ti konstante Terme oder Variablen ungleich X sind;

LGG (f(s1, ..., sn), f(t1, ..., tn)) = f (LGG(s1, t1), ..., LGG(sn, tn))

LGG (f(s1, ..., sn), g(t1, ..., tm)) = X

Beispiel

L1: unterhalt (ulf, maria, alimente(ulf, 1000))

L2: unterhalt(udo, marion, alimente(udo,300))

LGG(L1, L2): unterhalt (X, Y, alimente(X, V))

wobei es für jede andere Generalisierung G(L1, L2) eine Substitution gibt, so daß G(L1, L2) σ = LGG(L1, L2)

LGG (C1, C2)

C1: member(2, [1,2]) :- member (2, [2]).

C2: member (c, [a, b, c]):- member(c, [b,c]), member (c, [c]).

alle Paare:

[¬ m(2, [2]),   ¬ m(2, [2]), ¬m(2, [2]),   m(2, [1,2]),   m(2, [1,2]), m(2, [1,2])]

[¬ m(c, [b,c]), ¬ m((c, [c]), m(c, [a, b, c]), –m(c, [b,c]), –m((c, [c]), m(c, [a, b, c])]

LGG (¬ m(2, [2]), –m(c, [b,c])) = ¬m(A, [C|D])

LGG (¬ m(2,[2]), ¬ m(c, [c])) = ¬ m(A, [A])

LGG (m(2, [1,2]), m(c, [a, b, c])) = m (A, [B, C|D])

LGG (C1, C2) = m (A, [B, C|D]) :- m (A, [C|D]), m (A, [A]).

Bei jedem Literal probieren, ob es weggelassen werden kann, ohne zu generalisieren. Dieser Schritt ist leider NP-schwierig, weil der Subsumtionstest NP-schwierig ist.

Eigenschaften des LGG

kommutativ

lgg (e1, e2) = lgg (e2, e1)

assoziativ

lgg (lgg (e1, e2), e3) = lgg (e1, lgg (e2, e3))

idempotent

lgg (e, e) = e

Das bedeutet auch: reihenfolgeunabhängig

lgg (e1, e2, ..., en) = lgg (... lgg (lgg (e1, e2), e3), ... en)

und eindeutig.

In einem Verband von Äquivalenzklassen von Klauseln
ist das Supremum zweier Klauseln ihr LGG.

Aufwand

Die gute Nachricht:

Die Länge des LGG ist linear in der Anzahl der Selektionen.
Der Aufwand der Generalisierung ist linear in der Anzahl der Selektionen.

Die schlechte Nachricht:

Hat die längste Klausel in den Beispielen m Literale und
gibt es n Klauseln als positive Beispiele,
dann gibt es höchstens m^n Selektionen.

Es werden also exponentiell viele Selektionen in linearer Zeit bearbeitet.
Und dann kommt die Reduktion, die für jedes Literal noch einmal den aufwendigen Subsumtionstest braucht...