Sprachlernen: Positive Inferenz

Christian Fischbach

fischbac@ling.uni-duesseldorf.de

Wintersemester 1998/99

Seminar: Lernende Agenten im WWW

Dozentin: Prof. Dr. Katharina Morik

Lehrstuhl VIII: Künstliche Intelligenz

Fachbereich Informatik

Universität Dortmund

Schwerpunkt:

Angluin, Dana (1980b): *Inductive Inference of Formal Languages from Positive Data*, Information And Control 45, 117-135 (1980)

Hintergrund:

Angluin, Dana (1980a): *Finding Patterns Common to a Set of Strings*, Journal Of Computer And System Sciences 21, 46-62 (1980)

Gliederung

GTI flashback

Lerntheoretischer Rahmen

Positive Inferenz

(Nicht) inferierbare Sprachklassen - Beispiele

Vermeidung von Übergeneralisierung

Pattern Languages

GTI flashback - 1

Begriffe

Σ	festes endl. Alphabet aus Symbolen
Σ^*	Menge aller endlichen Wörter über Σ
$L \subseteq \Sigma^*$	Sprache (über Σ)
$\sigma = (s_i)$	unendl. Folge von Elementen $s_i \in \Sigma^*$ Positive Präsentation von L $:\Leftrightarrow$ $\{s_1, s_2, ...\} = L$
(L_i) mit $L_i \neq \emptyset$ und $\exists f$: $f(i,w) = \begin{cases} 1 & w \in L_i \\ 0 & sonst \end{cases}$	Indizierte Familie von nichtleeren rekursiven Sprachen

GTI flashback - 2

Chomsky-Hierarchie

Sprachklasse	*Automatenklasse*	*Wortproblem*
Rek. aufzählbar	TM	unentscheidbar
Rek. entscheidbar	haltende TM	entscheidbar
kontext-sensitiv	LBA	$O(a^n)$
kontext-frei	NPDA	$O(n^3)$
regulär	FA	$O(n)$

Lerntheoretischer Rahmen - 1

Identifikation im Grenzwert (IiG)

- Theoretisches *Lernparadigma* (*was* ist lernbar, unter *welchen* Umständen)
- Zugrundeliegende *Lernvorstellung*: Ermitteln einer Theorie / Funktion / Sprache anhand von Eingaben. Nach jeder Eingabe Ausgabe eines Lernergebnisses (Lernschritt)
- *Informale Definition*: Alg. *identifiziert das richtige Erg. im Grenzwert*, wenn, nachdem einmal (im Grenzwert) das richtige Lernergebnis gefunden wurde, nie wieder ein anderes gewählt wird.
  Formalere Def. folgt später über Inferenz-Machine (TM).

Lerntheoretischer Rahmen - 2

IiG-Paradigma vs. “ideales” Lernverfahren

- *“Ideales” Lernverfahren* (i. a. utopisch!)
  - *vollständig*, d.h. Ergebnis überdeckt alle positiven Beispiele,
  - *korrekt*, d.h Ergebnis überdeckt kein negatives Beispiel,
  - *terminiert* mit richtigem Ergebnis in *polynomieller Zeit*
- *Abstrich im IiG-Paradigma*: Verzicht auf Termination nach Finden des korrekten Ergebnisses

Lerntheoretischer Rahmen - 3

Lernbarkeit von Sprachklassen nach Gold (67)

Learnability model	*Class of Languages*
Anomalous text	Recursively enumerable recursive
Informant	Primitive recursive Context-sensitive Context-free Regular Superfinite
Text	Finite cardinality languages

Positive Inferenz - 1

Inferenzmachine (Motivation)

- *Formalisierung* von Sprachlernen (hier insbes. positiver Inferenz) im IiG-Paradigma
- intuitive *Arbeitsweise*: M liest die positiven Daten s_i der unendl. Eingabe $\sigma = (s_i)$ ein und produziert Hypothesenfolge $M[\sigma] = \tau$ (Folge von Gödelnummern)
- *Konvergenzbegriff* für $M[\sigma] = \tau$ als Definitionsgrundlage von positiver Inferenz
- Bestimmter *Typ von TM*; Abweichungen vom Standardmodell
  - aus Modellierungsgründen
  - keine Abschwächung der Mächtigkeit

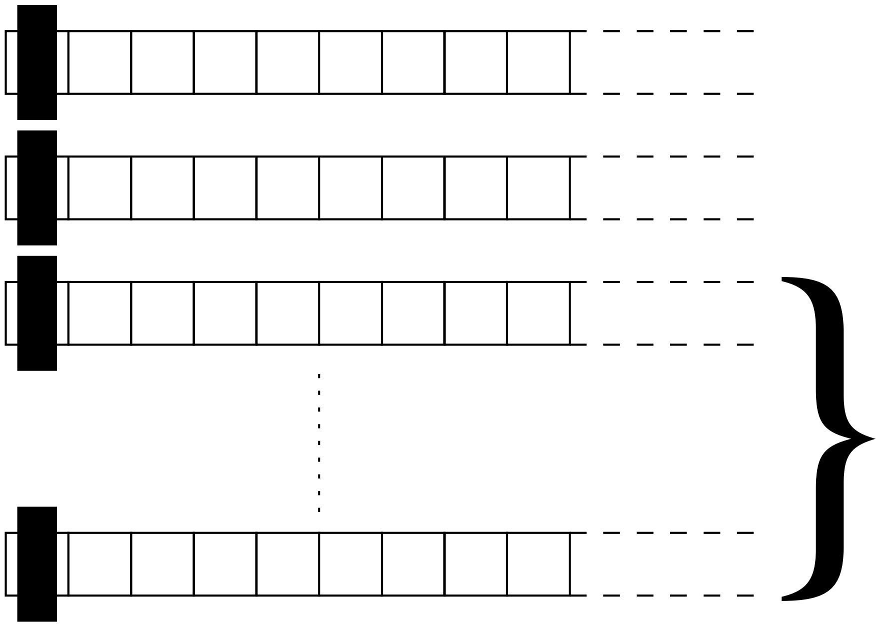
Positive Inferenz - 2

Inferenzmachine (formal & graphisch)

k-Band DTM $M = (Q, \Sigma, \Delta, B, q_0, \delta, F)$

Q={initial, request, answer, guess, ...},

$\delta : Q \times \Delta^k \rightarrow Q \times \Delta^k \times \{R, L, N\}^k$



sample-tape (Beispiele)

guess-tape (Hypothesen)

endlich viele
scratch-tapes
("Kritzel"-Bänder)

Positive Inferenz - 3

Inferenzmachine (prozedurales Schema)

unendl. **Eingabe:** $\sigma = (s_i)$, **Ausgabe:** $M[\sigma] = \tau$ Hypothesen

state = initial; $i = 0$; τ = null;
WHILE true DO {
i++;
SWITCH state {

"request"	$\Rightarrow$	$s_i B \rightarrow$ sample; head(sample) = 0; state = answer;
"guess"	$\Rightarrow$	$\tau(i)$ = gödel_nr(first_string(guess)); clear(guess);
		head(guess) =0;
X	$\Rightarrow$	gemäß δ

}};

Positive Inferenz - 4

Inferenzmachine (Konvergenz und pos. Inferenz)

$M[\sigma] = \tau$ *konvergiert gegen* $i \in \mathbb{N}$ $:\Leftrightarrow$
$\left(\; |\tau| = \infty \;\wedge\; \text{für fast alle } j \text{ ist } \tau_j = i \; \right) \;\vee$
$\left(\; 0 < |\tau| = j \;\wedge\; \tau_j = i \; \right)$

M *inferiert* L_i *aus positiven Daten* $:\Leftrightarrow$
$\forall \sigma = (s_k)$ von L_i $\exists j \in \mathbb{N} : M[\sigma]$ konvergiert gegen j und $L_j = L_i$

M *inferiert* (L_i) *aus positiven Daten* $:\Leftrightarrow$
$\forall i \geq 1 : \quad M$ inferiert L_i aus positven Daten

(L_i) *inferierbar aus positiven Daten* $:\Leftrightarrow$
$\exists M : \quad M$ inferiert (L_i) aus positiven Daten

Positive Inferenz - 5

Bedingungen in der Übersicht

Praktische Fragestellungen für konkrete Sprachklassen: Definition von positiver Inferenz über Inferenzmachine ziemlich unhandlich.

Herleitung von evtl. leichter abprüfbaren Bedingungen.

Formaler Zusammenhang in der Übersicht:

$$
\boxed{
\begin{array}{lllll}
\multicolumn{3}{l}{(L_i)\ \textit{positiv inferierbar}} & \Longleftrightarrow & (1)\ \textit{charakteristische Bedingung} \\
\Downarrow & \Uparrow & \Uparrow & & \\
(2) & (2)+(4) & (3) & & \textit{abgeleitete Bedingungen}\ \text{aus}\ (1)
\end{array}
}
$$

Positive Inferenz - 6

Charakterisierende Bedingung

Bedingung 1 *(aufzählbare verräterische Teilmengen)*
(L_i) *erfüllt Bed. 1* $:\Leftrightarrow\ \exists$ *Proz., die* $\forall i$ *eine Menge* T_i *aufzählt, so daß*
(i) $|T_i| < \infty$
(ii) $T_i \subseteq L_i$
(iii) $\forall j \geq 1 : T_i \subseteq L_j \Rightarrow L_j \not\subset L_i$

Intuitiv: T_i ist "verräterische" Teilmenge für L_i. Sobald T_i vollständig in Eingabe gelesen wurde, kann L_i ohne Gefahr von Übergeneralisierung ($\rightarrow$ (iii)) inferiert werden.

Theorem 1 (L_i) *inferierbar aus positiven Daten* $\Leftrightarrow$ (L_i) *erfüllt Bed. 1*

Positive Inferenz - 7

Exemplarischer Beweisansatz für "$\Leftarrow$" aus Theorem 1

Definition einer *schrittweisen Inferenzprozedur*, die (L_i) positiv inferiert und von einer Inferenzmachine implementiert werden kann.

> Schritt n: lese neues Bsp. s_n. $S_n = \{s_1, ..., s_n\}$.
>
> Wenn kleinstes $g \leq$ n mit $T_g^{(n)} \subseteq S_n \subseteq L_g$ existiert,
>
> dann Ausgabe Hypothese g.
>
> Gehe zu Schritt $n + 1$

Intuition: In Schritt n Hypothese g, wenn min. g ex., so daß

- L_g bisherige Eingabe S_n abdeckt und
- die ersten n aufgezählten Strings der "verräterischen" Teilmenge T_g bereits in Eingabe aufgetaucht sind

Positive Inferenz - 8

Variationen der charakterisierenden Bedingung

Forderung nach Aufzählbarkeit der $T_i \subseteq L_i$ ist essentiell!

- Verzicht auf diese Aufzählbarkeit (Bed. 2): dann Bedingung nicht mehr *hinreichend* für positive Inferenz
- Verstärkung der Forderung zu Entscheidbarkeit: dann Bedingung nicht mehr *notwendig* für positive Inferenz

Positive Inferenz - 9

Abgeleitete Bedingungen

Bedingung 2 *(schwache Bed. 1) Bed. 1 ohne Aufzählbarkeit der* T_i

Bedingung 3 *(endlich viele* L_i*-Obermengen für endl. Sprachen* S*)*

$\forall S \subseteq \Sigma^* \; mit \; 0 < |S| < \infty : für \; C(S) = \{L_i | S \subseteq L_i\} \; gilt \; |C(S)| < \infty$

Bedingung 4 *(berechenbare Inklusion)* $\exists h : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ *mit*

$$h(i, j) = \begin{cases} 1 & L_i \subseteq L_j \\ 0 & sonst \end{cases}$$

(Nicht) inferierbare Sprachklassen - Beispiele - 1

Unterklassen der regulären Sprachen

Reguläre Ausdrücke über {0,1} mit eingeschränkter Operatorenauswahl

Operatorenauswahl	*Positiv inferierbar?*	*Beweis*
$\cdot$ und *	nein	verletzt Bed. 2
$\cdot$ und $^+$	ja	erfüllt Bed. 3

Agluin(87): mit Membership- und Äquivalenz-Orakel ist RL lernbar (in der Praxis z.B. Datenbank als Orakel)

(Nicht) inferierbare Sprachklassen - Beispiele - 2

Pattern languages

Σ endl. Alphabet, $X = \{x_1, x_2, ...\}$ unendl. Menge von Variablen

Pattern p $\in (\Sigma \cup X)^+$. Es gilt: $0 < |p| < \infty$.

Sprache L(p) von p $L(p) = \{s \in \Sigma^+ | \; s \leq p \; \}$.

$s \leq p \quad :\Leftrightarrow \quad \exists \theta : X_p \rightarrow \Sigma^+ : s = p\theta$

Beispiel: $\Sigma = \{0, ...9\}, (\Sigma \cup X)^+ \ni p = 3x_6x_642x_{37}$. Dann z.B.
$3004213, 31221224255, 311421 \in L(p)$

Pattern languages $P_* = \{L(p) \; | \; p \text{ pattern } \}$

$P_i = \{L(p) \; | \; p \text{ pattern mit max. } i \text{ Variablen } \}$

(Nicht) inferierbare Sprachklassen - Beispiele - 3

Beweis: P_* positiv inferierbar

Erhalte $(L(p_i))$ durch systematische konkrete Aufzählung.

$(L(p_i))$ *ist positiv inferierbar*.

Beweis: $L(p_i)$ erfüllt Bed. 3 (hinreichend für positive Inferenz).
Z.z.: es ex. nur endl. viele $L(p_i)$-Obermengen für $S \subseteq \Sigma^*$.
$\forall S \subseteq \Sigma^*$ sei $l = min\{|s| \mid s \in S\}$. Dann gilt: $S \subseteq L(p) \Rightarrow |p| \leq l$.
Wegen $\left|\{L(p) \mid |p| \leq l\}\right| < \infty$ gilt die Behauptung.

Vermeidung von Übergeneralisierung - 1

Machinen-Eigenschaften

Intuitiv: M soll Übergeneralisierungen frühzeitig durch Konflikt mit bisher gelesenen Eingaben erkennen. M ist auf (L_i)

- *konsistent*: Hypothesen von M immer konsistent mit bisher eingelesen Beispielen
- *responsiv*: M antwortet mit mindestens einer Hypothese auf eine Eingabe bevor nächste Eingabe gelesen wird
- *konservativ*: M ändert Hypothese nur, wenn es einen Konflikt mit den bisher eingelesen Daten gibt

M ist konsistent nach Konstruktion, mit Modifikationen auch responsiv.

Vermeidung von Übergeneralisierung - 2

Hinreichende Bedingungen

Echte Einschränkung: *konservativ positiv inferierbare Sprachklassen* (es ex. pos. inferierbare (L_i), für die keine Inferenzmaschine konservativ sein kann)

Bedingung 5 *(Deskriptivität für endliche $\Sigma-$Sprachen)*
$\exists f : \Sigma^* \nrightarrow \mathbb{N}$, *für* $f(S) \mapsto k \;\; (0 < |S| < \infty)$ *ist* k *descriptiv für* S

k *descriptiv* für $S :\Leftrightarrow \quad S \subseteq L_k \;\; \wedge \;\; \forall j \geq 1 : S \subseteq L_j \Rightarrow L_j \not\subset L_k$
(L_k ist kleinste Obermenge für S bzgl. (L_i))

Intuition: Für gegebene Beispielmenge S ist L_k speziellste Hypothese

Theorem 2 (L_i) *erfüllt Bed. 3 & 5* $\Rightarrow \;\; \exists M$*, die* (L_i) *positiv inferiert und konsistent, responsiv und konservativ ist.*

Pattern languages - 1

Typische Entscheidungsprobleme & Abschlußeigenschaften

- *Membership* $s \in L(p)$? NP-vollständig! 1. in NP: rate Subst. σ, verifiziere $s = p\sigma$, 2. vollständig: 3-$SAT \leq_{pol} P_*$-membership (damit auch $p \leq q$ NP-vollständig).
- *Äquivalenz* $L(p) = L(q)$? Es gilt: $L(p) = L(q) \Leftrightarrow p \equiv q \Leftrightarrow \exists$ Var.-Umbenennung $f : f(q) = p$. Linear über $|p| = |q|$.
- *Teilmengenrelation* $L(p) \subseteq L(q)$? Entscheidbar für Spezialfälle $|p| = |q|$ oder $q \in P_1$. Allgemeiner Fall offen (1980).

P_* *nicht abgeschlossen* unter $\cup$,$\cap$, $()^C$, $()^+$,
abgeschlossen unter $\cdot$, $()^r$.

Pattern languages - 2

Inferenz von descriptiven Pattern - I

p *descriptiv* für $S :\Leftrightarrow \quad S \subseteq L(p) \quad \wedge \forall q$ mit $S \subseteq L(q) : \quad L(q) \not\subset L(p)$.
($L(p)$ minimal in Teilmengenordnung der S-enthaltenden Pattern Sprachen; $L(p)$ ist speziellste Hypothese für S bzgl. P_*).

Algorithmus (Eingabe: S, Ausgabe: p, descriptiv für S)
$l := min\{|s| \mid s \in S\}$. $C := \{p \mid |p| \leq l \quad \wedge \quad S \subseteq L(p)\}$.
$m := max\{|p| \mid p \in C\}$. $C' := \{p \in C \mid |p| = m\}$.
Ausgabe: $p \in C'$ mit $p \leq C'$.

Analyse: ineffizient.

S variabel, dann exponentiell in l.

S fest, dann NP-vollständige Tests in Berechnung (“$\leq$”).

Pattern languages - 3

Inferenz von descriptiven Pattern - II

Wegen offener Frage "$\subseteq$" löst Alg. härteres Problem mit Forderung "p hat max. Länge", es gilt: $|p| = |q| \;\Rightarrow\; (L(p) \subseteq L(q) \;\Leftrightarrow\; p \leq q)$

Theorem 3 *Wenn $P \neq NP$, dann ex. kein poly. Alg. für Problem "finde für gegebenes S descriptives p maximaler Länge".*

Beweisidee: Kontraposition. Es. ex. Alg. A. Mit Hilfe von A entscheide "$s \in L(p)$?" in poly. Zeit. Weil Membership NP-vollständig, dann $P = NP$.

Für "originales" Problem konnte Ergebnis nicht verifiziert werden ("$\subseteq$" offen)

Pattern languages - 4

Inferenz von descriptiven P_1-pattern - I

Theorem 4 *Es ex. poly.-zeit Alg. für das Problem "finde für gegebenes S eine P_1-pattern, das descriptiv für S bzgl. P_1 ist"*

Algorithmus basiert auf geschickter Zusammenschneidung von *P_1-Pattern-Automaten* (Repräsentation für Patterninduktion in Strings durch Teilstrings)

Formal: $A(s;t) = (Q, \Sigma, \delta, q_0, F)$, so daß
$L(A(s;t)) = P_1(s;t) = \{p \in P_1 \mid p[t/x] = s\}$.

$Q \ni (i, j)$, wenn i Σ-Symbole und j Variablen-Vorkommen bisher gelesen. $q_0 = (0, 0)$. $(i, j) \in F \Leftrightarrow i + j|t| = |s|$.

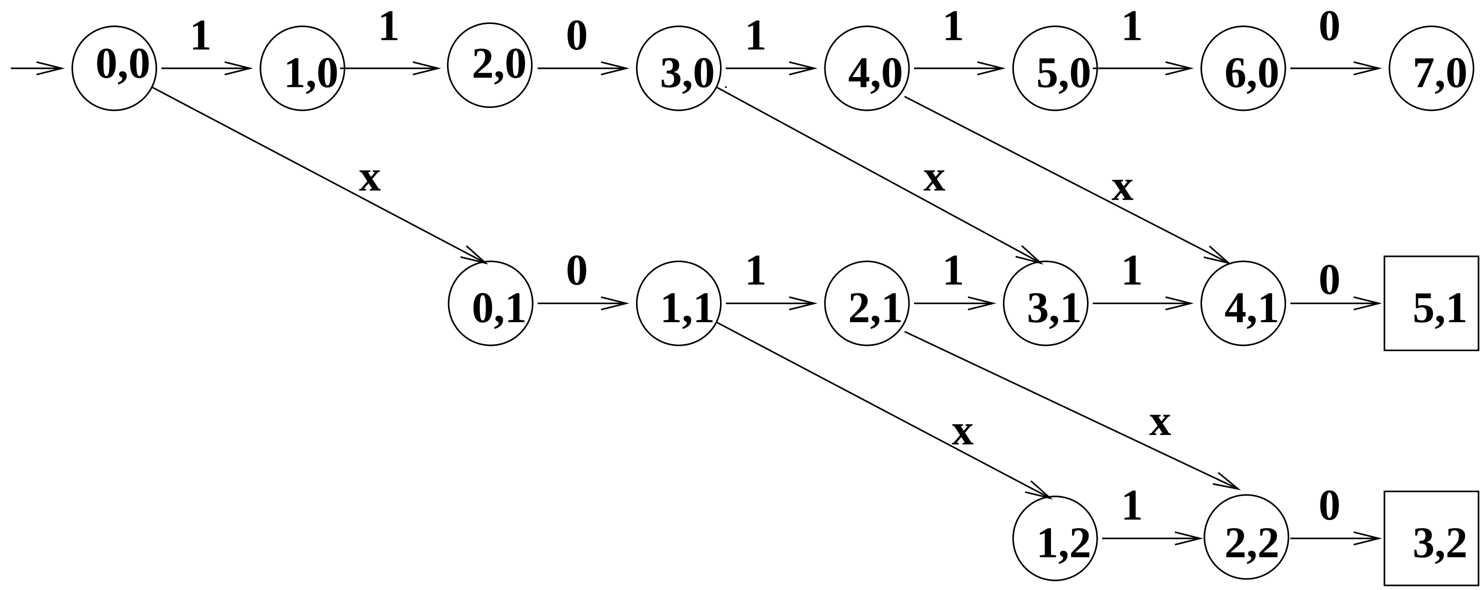
$\delta((i, j), b) = (i + 1, j)$ "Σ-Symbol b gelesen"
$\delta((i, j), x) = (i, j + 1)$ "Variable gelesen"

Pattern languages - 5

Inferenz von descriptiven P_1-pattern - II

Bsp.: $A(s;t)$ für $s = 1101110$, $t = 11$.

$L(A(s;t)) = \{x01110, 110x10, 1101x0, x0x10, x01x0\}$.



Pattern languages - 6

Definitionen zur Inferenz von descriptiven P_1-pattern

"Strukturtripel"

$\forall p \in P_1 : \quad \tau(p) = (i, j, k)$ wobei $i =\#$ Vorkommen $\Sigma-$Symbole, $j =\#$ Vorkommen Variablen, k am weitesten linke Pos. von x in p.

"Passende Strukturtripel"

$F_r = \{(i, j, k) \mid (i, j, k)$ paßt auf $s_r\}$

$F = \bigcap_{r=1}^{m} F_r$ (Triple, die auf alle Bsp. s_r $(1 \leq r \leq m)$ passen)

"P_1-Pattern-Automaten zu passenden Strukturtripeln"

$B_r(i, j, k)$ [aus $A_s(s_r; t)$] erzeugt pattern p, das s_r generiert und durch (i, j, k) beschrieben wird.

$B(i, j, k) = \bigcap_{r=1}^{m} B_r(i, j, k)$ erzeugt alle pattern p, die s_r $(1 \leq r \leq m)$ generieren und durch (i, j, k) beschrieben werden.

Pattern languages - 7

Algorithmus zur Inferenz von descriptiven P_1-pattern

Bestimme F.

$\forall (i,j,k) \in F$: Konstruiere $B(i,j,k)$.

Aus allen mögl. (i,j,k), für die $L(B(i,j,k)) \neq \emptyset$, wähle (i_0, j_0, k_0), so daß $i+j$ maximal.

Ausgabe: $p \in L(B(i_0, j_0, k_0))$.

Intuition: Alg. konstruiert Repräsentation aller P_1-pattern, die S generieren, partitioniert nach passenden Triplen (i,j,k).