

Lernalgorithmen

SoSe 2012 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Lernalgorithmen

Gesamtübersicht

0. Einführung
1. Identifikation (aus positiven Beispielen)
2. Zur Identifikation regulärer Sprachen, mit XML-Anwendung
3. HMM — Hidden Markov Models
4. Lernen mittels Anfragen & zur Roboterorientierung
5. Lernen mit negativen Beispielen
6. PAC-Lernen

Organisatorisches

Vorlesung DI 14.15-15.45, HZ 201

Publikum Hauptdiplomstudierende und Masterstudierende

Übungsbetrieb in Form von einer “Großen Übungsgruppe”

MI 8.25-9.55, F59

BEGINN: in der dritten Semesterwoche

Die VL in der zweiten SW wird verlegt auf die Übungszeit der Woche!

Dozentensprechstunde DO, 13-14 in meinem Büro H 410 (4. Stock)

Mitarbeitersprechstunde (Anna Kasprzik) DO 13-14 H 415

Modulprüfung: mündlich

Identifikation (aus positiven Beispielen): Literatur:

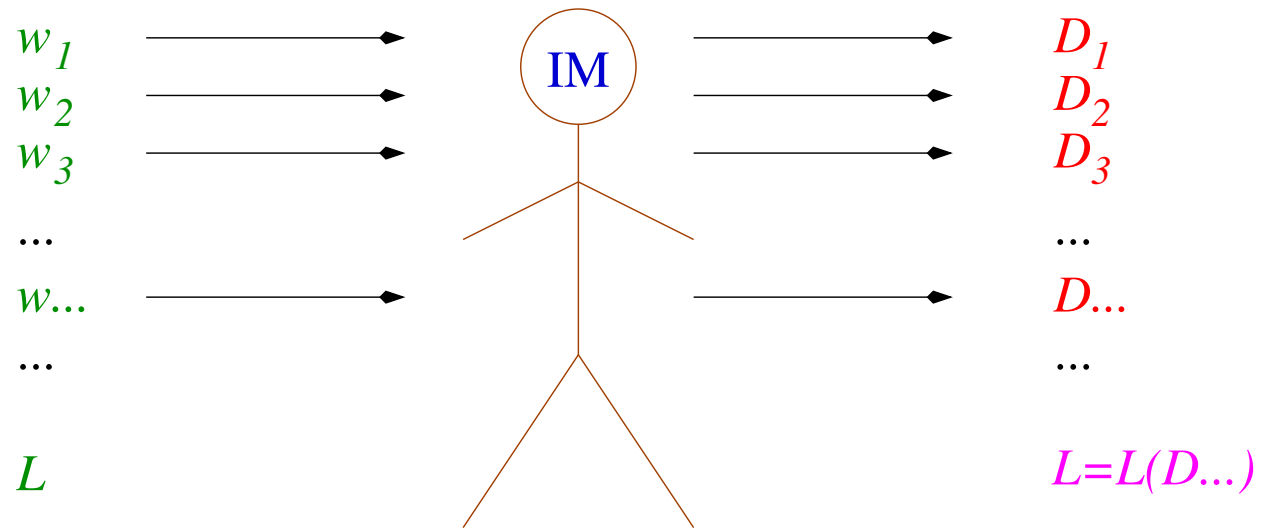
E.M. Gold. Language identification in the limit. *Information and Control* 10 (1967), pp. 447–474.

L. Blum, M. Blum. Toward a mathematical theory of inductive inference. *Information and Control* 28 (1975), pp. 125–155.

D. Angluin. Inductive inference of formal languages from positive data. *Information and Control* 45 (1980), pp. 117–135.

S. Jain, D. Osherson, J. S. Royer, A. Sharma. Systems that learn. 2. Auflage, MIT Press, 1999.

Golds Lernmodell des Textlernens



Mögliche Welten: Die Menge $\mathcal{E} \subseteq 2^{\mathbb{N}}$ der aufzählbaren “Sprachen”.

Zur Darstellung von *Beispielströmen*:

Ein *Text* sei irgendeine Abbildung $T : \mathbb{N} \mapsto \mathbb{N} \cup \{\#\}$.

$\#$: *Pausenzeichen*

Der *Inhalt* von T : $\text{Inh}(T) = \{T(i) \mid i \in \mathbb{N}, T(i) \neq \#\}$.

T heißt *Text für* $L \subset \mathbb{N}$, falls $\text{Inh}(T) = L$.

Hinweis: Für jede nicht-leere Sprache gibt es unendlich viele Texte, denn

(1) die Anordnung ist beliebig, (2) Wiederholungen sind zulässig, und

(3) $\#$ kann beliebig eingefügt werden.

Ein Text modelliere den Beispielstrom (aus der “gewählten” Sprache $L \in \mathcal{E}$) in den Lerner, der seine *Hypothesen* in Form von Turingmaschinen (Gödelnummern) oder Programmen oder Grammatiken / Automaten eines bestimmten Typs äußern kann.

Setze $T[n] = (T(0), \dots, T(n-1))$.

$SEQ = \{T[n] \mid T \text{ Text}, n \in \mathbb{N}\}$ ist die Menge aller endlichen Folgen über $\mathbb{N} \cup \{\#\}$.

Die *Länge* von $\sigma \in SEQ$ werde mit $|\sigma|$ bezeichnet.

Bem.: T beginnt mit σ genau dann, wenn $T[|\sigma|] = \sigma$.

Bem.: $T[n]$ ist die Datenmenge, die den Lerner bis zum n -ten Zeitpunkt zur Verfügung steht.

Gehen wir von einer fixierten Nummerierung aller Turingmaschinen aus, so können wir einen *Lerner*, auch *Inferenzmaschine (IM)* genannt, als (zunächst beliebige) partielle Funktion $F : SEQ \dashrightarrow \mathbb{N}$ verstehen.

Notation: $F(T[n]) \downarrow$: F ist definiert auf $T[n]$, das heißt, der Lerner äußert eine Hypothese nach Beobachtung von $(T(0), \dots, T(n-1))$.

Zugehörige Sprache: $L(F(T[n]))$.

F *konvergiert* auf Text T, falls es einen *Grenzwert* $i \in \mathbb{N}$ gibt,
so dass $\exists n_0 \in \mathbb{N} \forall n \geq n_0 : F(T[n]) = i$.
In Zeichen: $F(T) \downarrow = i$.

F *identifiziert* Text T genau dann, wenn
 $\exists i \in \mathbb{N} : F(T) \downarrow = i \wedge L(i) = \text{Inh}(T)$.

F *identifiziert* $L \in \mathcal{E}$ genau dann, wenn
 $\forall \text{Texte } T : \text{Inh}(T) = L \Rightarrow F \text{ identifiziert } T$.

F *identifiziert* $\mathcal{L} \subseteq \mathcal{E}$ genau dann, wenn
 $\forall L \in \mathcal{L} : F \text{ identifiziert } L$.

(sonst heit \mathcal{L} *nicht identifizierbar*.)

Anstelle von *Identifikation* spricht man auch vom *Textlernen*.

Die *Konkatenation* von $\sigma, \tau \in \text{SEQ}$ notieren wir durch Hintereinanderschreiben.
 $\sigma \sqsubseteq \tau$ ($\sigma \sqsubset \tau$) drücke die (echte) *Präfixrelation* aus.

Gilt $\sigma_0 \sqsubseteq \sigma_1 \sqsubseteq \dots \sqsubseteq \sigma_n \sqsubseteq \dots$, so bezeichne $\bigcup_{n \in \mathbb{N}} \sigma_n$ den dadurch eindeutig festgelegten Text.

Es seien $L \in \mathcal{E}$, ein Lerner F und $\sigma \in \text{SEQ}$ gegeben.

σ heißt *Schlussfolge* (locking sequence) für F auf L genau dann, wenn

(a) $\text{Inh}(\sigma) \subseteq L$,

(b) $L(F(\sigma)) = L$,

(c) $\forall \tau \in \text{SEQ}, \text{Inh}(\tau) \subseteq L \Rightarrow F(\sigma\tau) = F(\sigma)$.

Satz: (Blum & Blum) Identifiziert F eine Sprache $L \in \mathcal{E}$, so gibt es eine Schlussfolge für F auf L .

Beweis: Gegeben: $L \in \mathcal{E}$ und F . O.E. sei F auf ganz SEQ definiert.
Annahme: Es gibt keine Schlussfolge σ für F auf L . D.h.:

$$\begin{aligned} & \forall \sigma \in SEQ, \text{Inh}(\sigma) \subseteq L \wedge L(F(\sigma)) = L \\ \implies & \exists \tau \in SEQ : \text{Inh}(\tau) \subseteq L \wedge F(\sigma\tau) \neq F(\sigma). \quad (*) \end{aligned}$$

Wir zeigen: Aus $(*)$ folgt die Existenz eines nicht identifizierbaren Textes T für L .
Es sei $U = u(0), u(1), u(2), \dots$ ein beliebiger Text für L .
Wir definieren induktiv eine Folge $\sigma_n \sqsubset T$.

$n = 0 : \sigma_0 = \emptyset.$

$n \rightarrow n + 1 :$

1. Fall: $L(F(\sigma_n)) \neq L.$ Setze $\sigma_{n+1} = \sigma_n u(n).$

2. Fall: $L(F(\sigma_n)) = L.$

(*) gewährleistet die Existenz von $\tau \in \text{SEQ}, \text{Inh}(\tau) \subseteq L, F(\sigma_n \tau) \neq F(\sigma_n).$

Setze $\sigma_{n+1} = \sigma_n \tau u(n).$

Es ist $\sigma_i \sqsubseteq \sigma_{i+1}$ für alle i ; setze daher $T = \bigcup_{n \in \mathbb{N}} \sigma_n.$

Da U ein Text für L ist, ist auch T ein Text für $L.$

F konvergiert aber nicht auf T , denn für alle $n + 1 \in \mathbb{N}$ ist

entweder $L(\sigma_n) \neq L$

oder es gibt ein $\tau \in \text{SEQ}, \sigma_n \tau \sqsubseteq \sigma_{n+1}, F(\sigma_n \tau) \neq F(\sigma_n).$

□

Satz: (Angluin) $\mathcal{L} \subseteq \mathcal{E}$ ist identifizierbar genau dann, wenn
 $\forall L \in \mathcal{L} \exists D_L \subseteq L, |D_L| < \infty \forall L' \in \mathcal{L}, L' \neq L, D_L \subseteq L' \Rightarrow L' \not\subseteq L$.

Obiges D_L heißt auch *charakteristische (Teil-)Menge* von L .

Beweis: Es sei $\mathcal{L} \subseteq \mathcal{E}$ gegeben.

(1) Angenommen, F identifiziert \mathcal{L} .

Dann gibt es zu jedem $L \in \mathcal{L}$ eine Schlussfolge σ_L für F auf L .

Setze $D_L = \text{Inh}(\sigma_L)$.

Zu zeigen: $\forall L' \in \mathcal{L}, L' \neq L, D_L \subseteq L' \Rightarrow L' \not\subseteq L$.

Betrachte die Situation $D_L \subseteq L', L' \subseteq L$ für zwei $L, L' \in \mathcal{L}$.

Es sei T ein Text für L' mit $T[\|\sigma_L\|] = \sigma_L$ (wegen $\text{Inh}(\sigma_L) \subseteq L'$).

Da σ_L Schlussfolge, gilt: $L(F(\sigma_L)) = L$, also $L' = L(F(T)) = L$.

(2) Annahme: $\forall L \in \mathcal{L} \exists D_L \subseteq L (|D_L| < \infty) \forall L' \in \mathcal{L}, L' \neq L, D_L \subseteq L' \Rightarrow L' \notin \mathcal{L}. [+]$

Definiere einen Lerner F' wie folgt:

$F'(\sigma)$ sei das kleinste $i \in \mathbb{N}$, so dass für ein $L \in \mathcal{L}$ gilt:

(a) $L(i) = L$,

(b) $D_L \subseteq \text{Inh}(\sigma) \subseteq L$.

Gibt es solch ein i nicht, so sei F' undefiniert.

Zu zeigen: F' identifiziert \mathcal{L} .

Wir fixieren $L' \in \mathcal{L}$ und einen Text T für L' .

Es gibt dann ein $n_0 \in \mathbb{N}$, so dass $D_{L'} \subseteq \text{Inh}(T[n_0]) \subseteq L'$.

Sei i_0 der kleinste Index für L' . Nach Konstruktion von F' gilt:

F' konvergiert gegen i_0 auf T (und identifiziert so T), falls:

$\forall j < i_0, L = L(j) \in \mathcal{L} \forall m \in \mathbb{N}, D_L \subseteq \text{Inh}(T[m]) \exists n' > m \forall n \geq n' : \text{Inh}(T[n]) \not\subseteq L$.

Sei also $j < i_0$ und $m \in \mathbb{N}$ gegeben mit $L = L(j) \in \mathcal{L}, D_L \subseteq \text{Inh}(T[m])$.

Wegen $L \neq L'$ und $D_L \subseteq \text{Inh}(T) = L'$ gibt es wegen $[+]$ ein $x_0 \in L' \setminus L$.

T ist ein Text für L' , so dass es ein n' gibt mit $x_0 \in \text{Inh}(T[n])$ für alle $n \geq n'$.

Daher ist $\text{Inh}(T[n]) \not\subseteq L$.

□

Grenzen des Textlernens

Eine Sprachklasse, die alle endlichen sowie eine unendliche Sprache enthält, heißt *superfinit*.

Folgerung: (Gold) Keine superfinitive Sprachklasse ist identifizierbar.

Folgerung: Die Klasse der regulären Sprachen ist nicht identifizierbar.

Warum sind dies einfache Folgerungen ?

Hinweis: Im ersten Fall: Wähle als L vom Satz von Angluin eine unendliche Sprache.

Rekursive Identifizierbarkeit (d.h.: Lerner sind TM)

Satz: Es gibt eine identifizierbare, aber nicht rekursiv-identifizierbare Sprachfamilie.

Beweis: Es sei $K = \{n \in \mathbb{N} \mid \text{die } n\text{-te TM h\u00e4lt, angesetzt auf } n\}$ (*Halteproblemsprache*).

K kodiert das Halteproblem f\u00fcr Turingmaschinen und ist bekannterma\u00dfen rekursiv aufz\u00e4hlbar, aber nicht entscheidbar.

Betrachte $\mathcal{K} = \{K \cup \{x\} \mid x \in \mathbb{N}\}$.

\mathcal{K} ist identifizierbar durch

$$F(\sigma) = \begin{cases} \text{Index von } K, & \text{falls } \text{Inh}(\sigma) \subseteq K, \\ \text{Index von } K \cup \{x\}, & \text{falls } \exists x \notin K, x \in \text{Inh}(\sigma). \end{cases}$$

Wir zeigen, dass \mathcal{K} nicht rekursiv identifizierbar ist.

Angenommen, ein TM-berechenbarer Lerner F identifiziert \mathcal{K} .

Dann gibt es eine Schlussfolge σ_K f\u00fcr F auf $K \in \mathcal{K}$.

Fixiere eine berechenbare Aufz\u00e4hlung y_0, y_1, y_2, \dots von K .

F\u00fcr jedes $x \in \mathbb{N}$ bezeichne T^x den Text $\sigma_K x y_0 y_1 y_2 \dots$

Es gilt:

(a) $\forall x \in K : T^x$ ist Text für K .

(b) $\forall x \in \bar{K} : T^x$ ist Text für $K \cup \{x\} \neq K$.

Also: Nur Texte "für" \mathcal{K} !

Aus (a) folgt (denn σ_K ist Schlussfolge für K):

$\forall x \in K \forall n > |\sigma_K| : F(T^x[n]) = F(\sigma_K)$.

Aus (b) folgt (denn F identifiziert \mathcal{K}):

$\forall x \in \bar{K} \exists n > |\sigma_K| : F(T^x[n]) \downarrow \wedge F(T^x[n]) \neq F(\sigma_K)$.

Wäre F eine (partiell) rekursive Funktion, so auch

$$\chi : \mathbb{N} \dashrightarrow \mathbb{N}, \chi(x) = \begin{cases} \min\{S(n) \mid n > |\sigma_K| \wedge F(T^x[n]) \neq F(\sigma_K)\}, & \text{falls def.} \\ \uparrow & \text{sonst.} \end{cases}$$

Hierbei: $S(n)$ sei Schrittzahl einer F realisierenden TM bei Eingabe von $T^x[n]$.

Der Definitionsbereich von χ wäre \bar{K} , d.h., \bar{K} wäre rekursiv aufzählbar. Widerspruch! □

Satz: Es sei F^0, F^1, \dots eine Aufzählung aller Turing-berechenbaren (möglicherweise partiellen) Lerner.

Dann gibt es eine total-rekursive Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$, so dass

(a) $F^{f(i)}$ total ist und

(b) $\forall L \in \mathcal{E}$: falls F^i identifiziert L , dann gilt: $F^{f(i)}$ identifiziert L .

Beweis: $F^{f(i)}$ soll F^i simulieren, aber definiert sein auf σ , falls $F^i(\sigma) \uparrow$. Es sei $\sigma_{f(i)}$ das längste Präfix von $\sigma \in \text{SEQ}$ (falls dieses existiert), auf dem F^i nach $|\sigma|$ Schritten fertig ist.

$$F^{f(i)}(\sigma) = \begin{cases} F^i(\sigma_{f(i)}), & \text{falls } \sigma_{f(i)} \text{ definiert ist.} \\ 0, & \text{sonst.} \end{cases}$$

Sei T ein Text für L , und F^i identifiziere L , d.h., es gibt ein n_0 , so dass für alle $n \geq n_0$ $F^i(T[n]) = F^i(T[n_0])$ ein Index für L ist.

Sei s die Laufzeit zur Berechnung von $F^i(T[n_0])$.

$\forall n \geq \max\{n_0, s\} : F^{f(i)}(T[n]) = F^i(T[n_0])$, das heißt:

$F^{f(i)}$ konvergiert auf T gegen einen Index von L . □

Weitere Begriffe Speicherbeschränkte Lerner

Wir setzen:

$\sigma^- = \emptyset$, falls $\sigma = \emptyset$,

und $\sigma^- = (\sigma(0), \dots, \sigma(|\sigma| - 2))$, falls $\sigma \in \text{SEQ} \setminus \{\emptyset\}$;

$\sigma_{\text{last}} = \#$, falls $\sigma = \emptyset$,

und $\sigma_{\text{last}} = \sigma(|\sigma| - 1)$, falls $\sigma \in \text{SEQ} \setminus \{\emptyset\}$.

Ein Lerner F heißt *speicherbeschränkt*, falls

$\forall \sigma, \tau \in \text{SEQ} \ F(\sigma^-) = F(\tau^-) \wedge \sigma_{\text{last}} = \tau_{\text{last}} \Rightarrow F(\sigma) = F(\tau)$.

Mitteilung: Es gibt ein $\mathcal{L} \subseteq \mathcal{E}$, \mathcal{L} identifizierbar, aber nicht speicherbeschränkt identifizierbar.

Weitere Begriffe **Fette Texte**

Ein Text T heißt *fett*, falls

$$\forall i \in \text{Inh}(T) : |\{n \mid T(n) = i\}| = \infty.$$

Mitteilung: $\mathcal{L} \subseteq \mathcal{E}$ ist genau dann identifizierbar, wenn \mathcal{L} von einem speicherbeschränkten Lerner auf fetten Text identifiziert werden kann.

Weitere Begriffe **Konsistente Lerner**

Ein Lerner heißt *konsistent* auf L , falls $\forall \sigma, \text{Inh}(\sigma) \subseteq L \Rightarrow \text{Inh}(\sigma) \subseteq L(F(\sigma))$.

Satz: Jede konsistent rekursiv-identifizierbare Sprachfamilie $\mathcal{L} \subseteq \mathcal{E}$ enthält nur rekursive Sprachen.

Beweis: Es sei σ eine Schlussfolge für $L \in \mathcal{L}$.

Ist $x \in L$, so gilt $F(\sigma) = F(\sigma x)$.

Ist $x \notin L$, so ist —wegen der Konsistenz von F — $F(\sigma x)$ kein Index von L , da sonst $\text{Inh}(\sigma x) \not\subseteq L(F(\sigma x)) = L$ wäre; also ist $F(\sigma x) \neq F(\sigma)$.

Daher gilt: $x \in L$ genau dann, wenn $F(\sigma x) = F(\sigma)$.

F total rekursiv \leadsto Entscheidungsverfahren für L . □

Da $\{K\}$ trivial rekursiv identifizierbar ist (Erinner.: K ist die Halteproblemsprache), ist Konsistenz folglich eine echte Einschränkung. Es gilt sogar stärker:

Mitteilung: Es gibt auch $\mathcal{L} \subseteq \mathcal{E}$, die nur rekursive Sprachen enthalten, die rekursiv identifizierbar aber nicht konsistent rekursiv-identifizierbar sind.

Weitere Begriffe **Konservative Lerner**

Ein Lerner F heißt *konservativ* auf L genau dann, wenn $\forall \sigma \sqsubseteq \tau, \text{Inh}(\tau) \subseteq L \cap L(F(\sigma)) \Rightarrow F(\sigma) = F(\tau)$.

Mitteilung: Auch die Einschränkung auf konservative Lerner verkleinert die Familie der so rekursiv-identifizierbaren Sprachen.

Weitere Begriffe **Kleine Fehler**

L heißt *endliche Variante* von L' genau dann, wenn die symmetrische Differenz $L \Delta L' = L \setminus L' \cup L' \setminus L$ endlich ist.

$\mathcal{L} \subseteq \mathcal{E}$ *überdeckt* \mathcal{E} genau dann, wenn $\forall L \in \mathcal{E} \exists L' \in \mathcal{L}: L'$ ist endliche Variante von L .

Mitteilung: (Wiehagen): Es gibt eine rekursiv-identifizierbare Sprachfamilie $\mathcal{L} \subseteq \mathcal{E}$, die \mathcal{E} überdeckt.

Weitere Begriffe **Strenge Annäherung**

Fixiere eine Grammatikklasse \mathcal{G} . Eine Grammatik $G \in \mathcal{G}$ wird *streng angenähert*, wenn für die Hypothesenfolge $(G_t)_{t \in \mathbb{N}}$ zu jeder beliebigen Aufzählung von $L(G)$ gilt:

1. $\forall w \in L(G) \exists t_0 \in \mathbb{N} \forall t > t_0 : w \in L(G_t)$,
2. $\forall H \in \mathcal{G}, L(H) \setminus L(G) \neq \emptyset \exists t_0 \in \mathbb{N} \forall t > t_0 : G_t \neq H$ und
3. $\exists t_1 : L(G) = L(G_{t_1}) \wedge |\{t_j \mid G_{t_j} = G_{t_1}\}| = \infty$.

Mitteilung: (Feldman) Jede rekursive Grammatikklasse kann durch einen rekursiven Lerner streng angenähert werden.

Lernen von Funktionen

Texte enthalten Paare (x, y) , $x, y \in \mathbb{N}$, so dass in jedem Text G zu jedem $x \in \mathbb{N}$ genau ein $y \in \mathbb{N}$ erscheint, d.h. $(x, y) \in \text{Inh}(G)$.

G ist also ein Text für den Graph einer totalen Funktion $f : \mathbb{N} \rightarrow \mathbb{N}$.

SEG: die Menge aller endlichen Präfixe von Texten für Funktionen.

Den Begriff des Lernens im Limes kann man leicht auf Funktionen übertragen.

Satz: Die Menge \mathcal{R} der total-rekursiven Funktionen ist identifizierbar.

Beweis: Für $\sigma \in \text{SEG}$ sei $F(\sigma)$ der kleinste Index i einer totalen rekursiven Funktion φ_i mit $\text{Inh}(\sigma) \subseteq \text{Graph}(\varphi_i)$.

Da für alle $f, g \in \mathcal{R}$, $f \neq g$ es “Zeugen” in $x, y, z \in \mathbb{N}$ gibt mit $(x, y) \in \text{Graph}(f)$ und $(x, z) \in \text{Graph}(g)$, $y \neq z$, folgt die Limeslerneigenschaft von F . \square

Mitteilung: (Gold) \mathcal{R} ist nicht rekursiv-identifizierbar.

Zum weiteren Vorlesungsablauf

Bislang: Lerntheorie

Hypothesen sind meist Zahlen (Gödelnummern)

Weiteres siehe Spezialvorlesung Lern+Rekursionstheorie im WS

Danach / ab der nächsten Vorlesung: Grammatische Inferenz

Hypothesen sind konkrete Grammatiken / Automaten etc.

⇒ Die speziellen Eigenschaften solcher Sprachbeschreibungen können ausgenutzt werden.

⇒ Wir lernen Algorithmen kennen, die programmiert und auch praktisch eingesetzt werden können.