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 Hauptdiplomsstudierende 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

Probleme ? Fragen ?

Klären Sie bitte Schwierigkeiten mit Vorlesungen oder Übungen möglichst umgehend in den zur Verfügung gestellten Sprechzeiten.

Wir helfen Ihnen gerne!
. . . wir sind aber keine Hellseher, die Ihnen Ihre Schwierigkeiten an der Nasenspitze ansehen. . .

Einführung 1: Einordnung

Was ist Lernen?

Eine Aufgabe für Psychologen, Informatiker, Biologen, Philosophen, ...

Eine naturwissenschaftliche Sicht

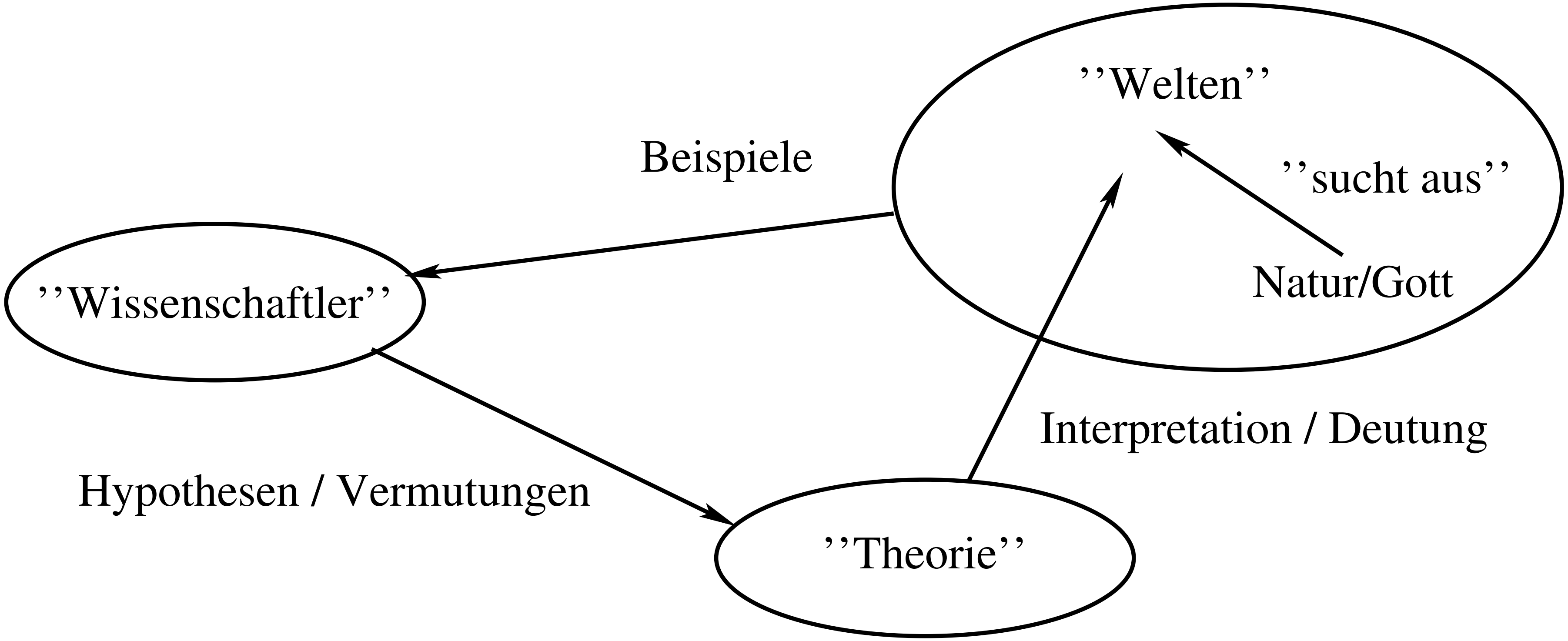
Induktion versus Deduktion

In der Physik gibt es zwei unterschiedliche, sich ergänzenden Vorgehensweisen:

(A) die Herleitung von allgemeinen Gesetzmäßigkeiten aus durch Beobachtung oder Experiment gewonnenen Beispieldaten. [Induktion]

(B) das logische Schlussfolgern von Gesetzmäßigkeiten aus postulierten Grundgesetzen (Axiomen). [Deduktion]

Grundschema des Lernens:



Konzeptlernen als **informatische Aufgabe**

Formalisiert sehen wir als Aufgabe des (maschinellen) Lernens, Konzepte zu ergründen, wobei wir uns auf Konzeptklassen $\mathcal{C} \subseteq 2^{\mathbb{N}}$ oder $\subseteq \mathbb{N}^{\mathbb{N}}$ einschränken:

Beispiel 1: $\mathcal{C}_1 = \{M_n \subseteq \mathbb{N} \mid n \in \mathbb{N},\ M_n = \mathbb{N} \setminus \{n\}\}$.

Beispiel 2: $\mathcal{C}_2 = \mathcal{C}_1 \cup \{\mathbb{N}\}$.

Beispiel 3: $\mathcal{C}_3 = \{f : \mathbb{N} \to \mathbb{N} \mid f = id \vee$
$(\exists n \in \mathbb{N} : f|_{\{0,\ldots,n\}} = id \wedge \forall\, m > n : f(m) = m + 1)\}$.

Bemerkung: $\mathcal{C}_3 \triangleq \mathcal{C}_2$ mit Sortierung.

Präsentation des Konzepts

(i) Eine Turingmaschine zählt das gewählte Konzept $C \in \mathcal{C}$ auf, bei Mengen eventuell mit Wiederholungen;
in Beispiel 3 in der Reihenfolge der Funktionswerte $f(0)$, $f(1)$, $f(2)$, ... ;
bei Konzepten $C \subseteq \mathbb{N}$ (oder $C \subseteq \Sigma^*$) auch paarweise $(c, t) \in C \times \{0, 1\}$, wobei $c \in C \Leftrightarrow (c, 1)$ aufgezählt wird und $c \notin C \Leftrightarrow (c, 0)$ aufgezählt wird.

(ii) Der Wissenschaftler / Lerner kann sich gezielt Informationen beschaffen (z.B. durch Experimente).
Dies kann zur Definition von 2-Personenspielen führen, wobei die Frage der Lernbarkeit zur Frage nach einer Gewinnstrategie wird.

Klassifikation und Adaption

Anwendungenrelevante Aufgabe: Eingaben automatisch zu *klassifizieren*.
Dazu nützliche Herangehensweise:
Erlerne Klassifikationsschema in oben skizzierter Weise aus Beispielen;
$\rightsquigarrow$ *Lernphase*
dieses wird — quasi in einer zweiten Phase — als *Klassifikator* benutzt.
$\rightsquigarrow$ *Anwendungsphase*
Typische Vertreter dieses Zwei-Phasen-Schemas sind *neurale Netze*.
In gewissem Sinne fallen hierunter auch *adaptive Algorithmen*.
Bei ihnen wechseln sich Lern- und Anwendungsphase stets ab, wodurch es zu *Rückkopplungen* (mit evtl. Schwingungen) kommt.

Kindliches Lernen, z.B. Spracherwerb

Lehrer (?)

Aspekte des Spracherwerbs

Phonetik: Lautkunde

Lexik(on):

Wortschatz / Wortbildung

Grammatik

(Pragmatik)

Informatik ?

Mathematik ?

Technikbezug ?!

Kindlicher Spracherwerb: manchmal aktiver

Modellierung?!
$\leadsto$ Betrachtung unterschiedlicher Lernszenarien
insbesondere:
—unterschiedliche Rollen von Lehrer und Schüler,
—soziale Gefüge (Protokolle)

Autonomes System: Antwort auf sich wandelnde Umgebung

$\rightsquigarrow$ Notwendigkeit einer formalisierten Sicht des Lernens

$\boxed{\textbf{“Intelligenztest”}}$ spielerische Annäherung ans Sprachenlernen

Frage: Welche Sprache ist (genau) beschrieben durch:

$$\{ab, abb, abbb, abbbb, abbbbb, \ldots\}?$$

Was ist die Struktur dieser Sprache ?

Begriffserklärungen:

Sprache: Menge von Wörtern

Wort: Aneinanderreihung von Zeichen

mathematischer: Element eines frei erzeugten Monoids

Sprachklasse: Menge von Sprachen

$\boxed{\text{“Intelligenztest”}}$

Welche Sprache ist (genau) beschrieben durch:

$$\{ab, abb, abbb, abbbb, abbbbb, \ldots\}?$$

Antwort: ab^+ (?)

D.h.: Erst kommt ein a, dann beliebig viele b’s, mindestens ein b.

“Intelligenztest”

Welche Sprache ist (genau) beschrieben durch:

$$\{ab, abb, abbb, abbbb, abbbbb, \ldots\}?$$

andere Antwort: $\{ab^{10n+k} \mid n \geq 0, k = 1, 2, 3, 4, 5\}$ (?)
Also: Das “nächste” Wort wäre $abbbbbbbbbbb = ab^{11}$.
Welche Lösung ist “richtig” ?

$\boxed{\textbf{Einordnung der Vorlesung:}}$ Zusammenhänge zu anderen Gebieten

- Statistik, insbesondere Parameterschätzung ($\longrightarrow$ Mathematik)
- Numerik, insbesondere Interpolation ($\longrightarrow$ Mathematik)
- adaptive Algorithmen
- Kryptographie, Datenkompression
- andere Teilaspekte maschinellen Lernens, z.B.: neurale Netze, fallbasiertes Schließen, Lerntheorie
  (Hinweis: Vorlesung Lerntheorie und Rekursionstheorie im kommenden WS)

Einordnung der Vorlesung: “Hilfsmittel” theoretischer Untersuchungen:

- Rekursionstheorie / Berechenbarkeit
- Komplexitätstheorie / Algorithmik
- Automatentheorie / Formale Sprachen
- (Kolmogorov-Komplexität / Informationstheorie)

Einführung 2: Literatur

Das Buch zu dem Thema:

Grammatical Inference, Learning Automata and Grammars
(ISBN-13: 9780521763165)
by Colin de la Higuera
Cambridge University Press, 2010.

Weitere Bücher und Skripten

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

M. J. Kearns und U. V. Vazirani. An Introduction to Computational Learning Theory. 2. Auflage, MIT Press, 1997.

P. Langley. Elements of Machine Learning. Morgan Kaufmann Publishers, 1996.

T. M. Mitchell. Machine Learning. McGraw-Hill, 1997.

B. K. Natarajan. Machine Learning; a Theoretical Approach. Morgan Kaufmann Publishers, 1991.

K. Morik. Maschinelles Lernen; Skript zur Vorlesung, 12. April 1999; Universität Dortmund.
`http://www-ai.cs.uni-dortmund.de/LEHRE/VORLESUNGEN/MLRN/mlrn.h`

F. Stephan. Einführung in die Lerntheorie.
`http://math.uni-heidelberg.de/logic/fstephan/fstephan.html`

F. Wysotski. Skript zur Vorlesung Maschinelles Lernen, Wintersemester 99/00; Institut für Angewandte Informatik der TU Berlin.
`http://ki.cs.tu-berlin.de/lehre/ml.html`

Lern-Konferenzen

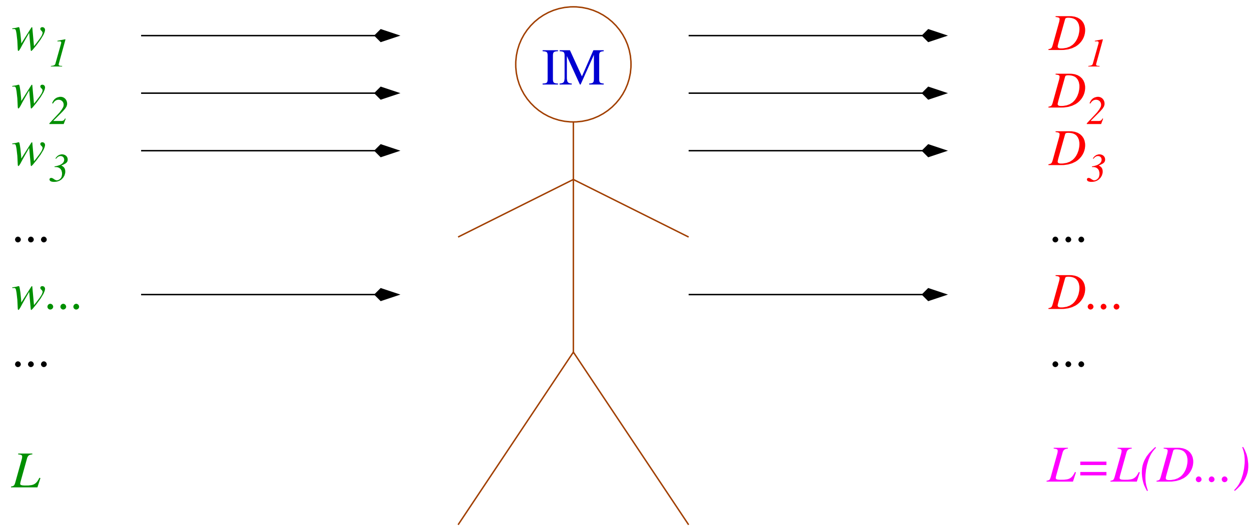
- International Conference on Machine Learning (ICML)
- European Conference on Machine Learning (ECML)
- Conference on Learning Theory (CoLT)
- [European Conference on Learning Theory (EuroCoLT)]
- Algorithmic Learning Theory (ALT)

- International Conference on Grammatical Inference (ICGI)
- Principles of Data Mining and Knowledge Discovery (PKDD)
- Discovery Science (DS)
- International Conference on Pattern Recognition (ICPR)
- Syntactical and Structural Pattern Recognition (SSPR)
- ...

Lernalgorithmen: Ein Schwerpunktthema des Lehrstuhls, z.B. 2012
$\Rightarrow$ Einladung für Seminar- / Master- / Diplom- / etc. Arbeiten

- H. Fernau. Approximate learning vs. inductive learning. Encyclopedia of the Sciences of Learning
- A. Kasprzik. One-shot learning and the polynomial characterizability barrier. TR 12-4 (Uni Trier).
- J. Björklund, H. Fernau, A. Kasprzik. Polynomial inference of universal automata from membership and equivalence queries. TR 12-3 (Uni Trier).
- A. Kasprzik. Formal tree languages and their algorithmic learnability. (Dissertation)
- H. Fernau: PC Mitglied ICGI 2012

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: $\mathrm{Inh}(T) = \{T(i) | i \in \mathbb{N}, T(i) \neq \#\}$.

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

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 $T[n] = (T(0), \ldots, T(n-1))$.
$SEQ = \{T[n] | T \text{ Text}, n \in \mathbb{N}\}$ 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 $F : SEQ \multimap \mathbb{N}$ verstehen.
Notation: $F(T[n]) \downarrow$: F ist definert auf $T[n]$, das heißt, der Lerner äußert eine Hypothese nach Beobachtung von $(T(0), \ldots, T(n-1))$.
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
$\exists i \in \mathbb{N} : F(T) \downarrow= i \wedge L(i) = Inh(T)$.

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

F *identifiziert* $\mathcal{L} \subseteq \mathcal{E}$ genau dann, wenn
$\forall L \in \mathcal{L} : F$ identifiziert L.
(sonst heißt $\mathcal{L}$ *nicht identifizierbar*.)
Anstelle von *Identifikation* spricht man auch vom *Textlernen*.

Zurück zu den Beispielen

$\mathcal{C}_1 = \{M_n \subseteq \mathbb{N} \,|\, n \in \mathbb{N},\ M_n = \mathbb{N} \setminus \{n\}\}$.

Ein Text für M_2 könnte sein: $5\#\#3\#3\#1\#(11)\#0\#00(13)\cdots$ Klammern als Hilfssymbole

Da klar ist, dass $\mathcal{C}_1$ die Konzeptklasse ist, wäre n eine sinnvolle Codierung der Hypothese M_n.
Welche Hypothese sollte ein Lerner nach Empfangen von $T[n]$ äußern?

Klappt das auch für $\mathcal{C}_2 = \mathcal{C}_1 \cup \{\mathbb{N}\}$?