GRUNDLAGEN ALGORITHMISCHER INFORMATIONSTHEORIE

(soweit für's maschinelle Lernen relevant)
VORLÄUFIGE, STICHWORTARTIGE BEILAGE ZUR VORLESUNG **Maschinelles Lernen II (SS 2005)**
Zusätzliches Material zu Solomonoff-Induktion und universellen Lernmaschinen auf der WWW-Vorlesungsseite:
`http://www.idsia.ch/~juergen/mlbio2.html`

Prof. Dr. habil. Jürgen Schmidhuber
Fakultät für Informatik
TUM

Kapitel 1

ÜBERBLICK

- ZEIT/ORT: Freitag, 10.00-12.00
- GEBIET: ... prüfbare Vorlesung
- HÖRERKREIS: nach DVP, Ausnahmen möglich
- VORAUSSETZUNGEN: DVP Wissen; Material aus der Vorlesung Maschinelles Lernen I; etwas Algorithmen und Berechenbarkeit/Logik ist von Vorteil. Voraussichtliche Bedingung zur Vergabe eines benoteten Scheins: mündliche Prüfung.
- GEEIGNET einerseits für Studienschwerpunkt Algorithmen und Berechenbarkeit/Logik, besonders aber auch für Fopras, Hauptseminare und Diplomarbeiten im Bereich des maschinellen Lernens.

WARNUNG: *Dieses Skript bietet Ihnen lediglich ein skelettartiges Gerüst eines Teils des Vorlesungsstoffes, nämlich desjenigen Teils, der für's maschinelle Lernen relevante Grundlagen der algorithmischen Informationstheorie abdeckt. mit dem Sie Ihre eigene, hoffentlich detailliertere Mitschrift vergleichen können. Bitte überprüfen Sie selbst noch einmal alle mathematischen Herleitungen. Der Dozent freut sich über jeden Fehler, den Sie entdecken. Gute Begleitlektüre: [22, 12]. In der Vorlesung wurden auch folgende separate Arbeiten behandelt: [35, 34].*

1.1 INHALT

Die Kolmogorov Komplexität eines berechenbaren Objektes ist die Länge des kürzesten Programms, das das Objekt berechnet. Sie ist im wesentlichen unabhängig von der Maschine, die das Programm ausführt. Konzepte der Theorie der Kolmogorov Komplexität stellen Informationstheorie und Wahrscheinlichkeitstheorie auf eine neue Grundlage. Sie finden mehr und mehr Verwendung in den verschiedensten Bereichen, u.a. auch dem maschinellen Lernen. Subziel der Vorlesung ist das Verständnis der wesentlichen für ML relevanten Grundlagen sowie einiger bedeutender Anwendungen.

- Kolmogorov Komplexität
  a) Überblick, Einführung von Turing Maschinen etc.
  b) Definitionen und Varianten
  c) Invarianztheorem
  d) Halteproblem und Unberechenbarkeit der Kolmogorov Komplexität
- Algorithmische Wahrscheinlichkeit
  a) Probleme der konventionellen Wahrscheinlichkeitstheorie

b) Solomonoff-Levin Verteilung

c) Algorithmische Entropie

d) Dominanz der kürzesten Programme

- Algorithmische Informationstheorie

  a) Einführung in relevante Konzepte der Informationstheorie: Entropie, Kullback-Leiber Distanz, wechselseitige Information.

  b) Bedingte Kolmogorov Komplexität, bedingte algorithmische Information, Unterschiede zur klassischen Informationstheorie.

- Was ist Generalisierungsfähigkeit? (Hierzu gibt's zusätzliches, separates, weiterführendes Material, insbesondere Hutter's Folien)

  a) Fundamentale Schranken des Lernbaren

  b) Ockhams Rasiermesser: bevorzuge einfache Datenmodelle.

  c) Solomonoffs Theorie der induktiven Inferenz

  d) Das Prinzip der minimalen Beschreibungslänge

- Kolmogorov-basierte Such- und Lernalgorithmen. (Hierzu gibt's zusätzliches separates, weiterführendes Material)

  a) Levin Komplexität

  b) Der optimale universelle Suchalgorithmus und seine inkrementellen Erweiterungen

  c) Maschinelle Lernverfahren zur Bestrafung überkomplexer Datenmodelle. Anwendungen für neuronale Netze.

  d) Mögliche Erweiterungen für inkrementelles Lernen

Kapitel 2

EINFÜHRUNG

"We are to admit no more causes of natural things (as we are told by Newton) than such as are both true and sufficient to explain their appearances. This central theme is basic to the pursuit of science, and goes back to the principle known as Occam's razor: "if presented with a choice between indifferent alternatives, then one ought to select the simplest one". Unconsciously or explicitly, informal application of this principle in science and mathematics abound." (Li and Vitányi in [21].)

2.1 GRUNDLAGEN

- Die erste Zahl ist gleich 2. Die zweite Zahl ist gleich 4. Die dritte Zahl ist gleich 6. Die vierte Zahl ist gleich 8. Wie lautet die fünfte Zahl? Sie lautet 34, denn die $n$te Zahl ist gleich

$$n^4 - 10n^3 + 35n^2 - 48n + 24.$$

  Aber IQ-Test verlangt Antwort "10". "Simple" Lösungen bevorzugt! Aber was heißt "simpel"?

- Im folgenden: Simplizität bzw. Komplexität berechenbarer Objekte definiert über die Länge der kürzesten Programme, die die Objekte berechnen.

- Zunächst: Turing Maschine (TM) mit Eingabeband, Rechenband, Ausgabeband. Lesekopf für jedes Band. Eingabekopf nur nach rechts rückbar. Ausgabekopf nur nach rechts rückbar. Rechenkopf in beide Richtungen rückbar. Alphabet für Eingabeband: $EA = \{0, 1\}$. Alphabet für Rechenband: $RA = \{0, 1, \_\}$. Alphabet für Ausgabeband: $AA = \{0, 1\}$. TM hat n Zustände $Z_1, ..., Z_n$. Z_1 Anfangszustand. Kann m Aktionen ausführen: $A_1, ..., A_m$. Menge aller Zustände: Z. Menge aller Aktionen: A. $n \times 3$ Tabelle bildet Paar $(z, c) \in Z \times RA$ auf ein Paar $(a, z) \in A \times Z$ ab (c ist Zeichen auf dem Feld über dem Rechenkopf). $m = 9$ Aktionen:

  1. Halt. 2. Rücke Rechenkopf ein Feld nach rechts. 3. Rücke Rechenkopf ein Feld nach links. 4. Schreibe '_' auf Feld über Rechenkopf. 5. Schreibe '0' auf Feld über Rechenkopf. 6. Schreibe '1' auf Feld über Rechenkopf. 7. Mach' Zeichen über Rechenkopf gleich dem Zeichen über Eingabekopf und rücke Eingabekopf ein Feld nach rechts. 8. Mach' Zeichen über Ausgabekopf gleich '1' und rücke Ausgabekopf ein Feld nach rechts. 9. Mach' Zeichen über Ausgabekopf gleich '0' und rücke Ausgabekopf ein Feld nach rechts.

- TM C: partielle Fn.: $f_C : \{0, 1\}^* \to \{0, 1\}^*$. Undefiniert, wo C nicht hält.

- **Selbstbeschränkende Programme.** Sei $| s |$:= Zahl der bits im bitstring s. p selbstb. <> U liest alle $| p |$ bits und hält (trägt also Information über seine eigene Länge). Kein selbstb. Prog. Prefix eines anderen [43] [18] [4]!

- **Compilertheorem.** Bekannt: Es ex. univ. TM U so daß f.a. TM C ex. $\mu_C : f_C(p) = f_U(\mu_C p)$ f.a. Programme p, wobei μ_C konstanter Prefix (Compiler).
- **Kolmogorov Komplexität** (auch algorithmische Komplexität, alg. Info). "Moderne" Fassung:
$$K_U(s) = min\{\mid p \mid\mid p \; selbstb., \; f_U(p) = s\}.$$
(Frühere Fassung: es wurde kein Wert auf selbstbeschränkende Programme gelegt.)
- **O-Schreibweise**.
$f(x) = O(g(x))$ falls positive Konstanten c, x_0 existieren, so daß $\mid f(x) \mid \leq c \mid (g(x)) \mid$ für alle $x \geq x_0$.
- **Invarianztheorem**. $K_{U_1}(s) = K_{U_2}(s) + O(1)$ für 2 univ. TM U_1 and U_2. Warum? Wegen **Compilertheorem.**
Ab jetzt: wir wählen eine universelle TM U und schreiben $K(s) = K_U(s)$.

2.1.1 WIEVIEL IST KOMPRIMIERBAR?

- **Zufällige Zeichenketten**: Solche s, deren kürzestes Programm nicht wesentlich kürzer als s ist: $K(s) \sim \mid s \mid$.
- **Regelmäßige Zeichenketten**: Solche s, deren kürzestes Programm wesentlich kürzer als s ist: $K(s) << \mid s \mid$.
- **Fast alle Zeichenketten sind unregelmäßig**: Zahl der bitstrings mit n oder weniger bits: $2^{n+1} - 1$. Zahl möglicher Programme mit weniger als n bits: $< 2^n$. Zahl möglicher Programme mit weniger als $n - k$ bits: $< 2^{n-k}$. Höchstens ein Tausendstel aller bitstrings der Länge 1000000 lassen sich durch ein Programm berechnen, welches weniger als 999990 bits umfaßt.

2.1.2 UNBERECHENBARKEIT DER KOLMOGOROV KOMPLEXITÄT

I.a. ist $K(s)$ nicht berechenbar – das kürzeste Prog. läßt sich nicht finden.

Dazu: Berry's Paradox: "Finde die kürzeste Zahl, deren Beschreibung mehr Zeichen benötigt, als in diesem Satz sind."

Oder: "Finde die kürzeste Zahl, deren Beschreibung weniger als 1000000 Zeichen benötigt."

Die kürzeste Beschreibung hat also mehr als 1000000 Zeichen. Aber die Zahl läßt sich doch offensichtlich mit weniger als 30 Zeichen beschreiben? Paradox.

Man muss formaler werden. Betrachte den ersten Bitstring, **von dem bewiesen werden kann**, daß seine Komplexität 1000000 Bits übersteigt.

Dazu: Schreibe Programm, daß bei gegebener formaler Sprache mit Menge von Axiomen und Schlußregeln alle Theoreme geordnet nach Beweislänge ausgibt. Irgendwann wird jedes Theorem ausgedruckt werden.

Betrachte nun "den ersten String s mit Beweis von $(K(s) > 1000000)$". Kann durch kurzes Programm (fixer Länge) gefunden werden, das systematisch alle Beweise auflistet, bis es s findet (Programmeingabe ist 1000000 - kann durch $log 1000000$ Bits dargestellt werden).

$logn + c$ Bits reichen also, den ersten String s mit Beweis von (K(s) ¿ n) zu berechnen. Widerspruch, da $logn + c$ viel langsamer wächst als n!

Für groß genuge n kann also der erste String s mit Beweis von $(K(s) > n)$ nicht existieren! Für groß genuge n kann nicht bewiesen werden, daß die Komplexität eines strings n übersteigt, obwohl die die Komplexität fast aller strings n übersteigt!

Kapitel 3

ALGORITHMISCHE WAHRSCHEINLICHKEIT

Hierzu gibt's zusätzliches, separates, in der Vorlesung besprochenes, weiterführendes Material, insbesondere Hutter's Folien. Hier nur ein grober Überblick über ein paar relevante Konzepte.

Axiomatische Wahrscheinlichkeitstheorie: [13]. Jeder verwendet sie. Aber seit langem ärgert man sich über das Problem des Bezugs zur realen Welt und zu relativen Häufigkeiten. Beispiel: Ereignis A habe best. Wahrsch. Dann: "Bei n Versuchen liegt die Zahl der Ereignisse A mit Wahrscheinlichkeit P zwischen zwei Zahlen a und b". Schon von Mises [40] kritisiert zirkuläre Argumentation.

3.1 SOLOMONOFF-LEVIN VERTEILUNG

Im folgenden: Programme halten, sonst sind sie keine.

$P_U(s)$, die *a priori Wahrsch.* des bitstrings s, ist die Wahrsch., ein (haltendes) Programm für U zu raten, welches s berechnet.

Dabei "raten" wie folgt definiert:

TM wie im Kapitel 2. Unterschied: Zu Beginn der Rechnung (TM im Initialzustand) ist Eingabeband leer. Wann immer Eingabekopf nach rechts rückt, *würfle* das nächste Bit: Mit Wahrsch. $\frac{1}{2}$ schreibe '1' in das Feld über Eingabekopf. Mit Wahrsch. $\frac{1}{2}$ schreibe '0'.

- Wahrscheinlichkeit, bestimmtes Programm p zu würfeln, ist $\frac{1}{2}^{|p|} = 2^{-|p|}$.
- Summe aller dieser Wahrscheinlichkeiten kann 1 nicht übersteigen (kein haltendes Programm ist Prefix eines anderen).

Algorithmische Wahrscheinlichkeit der Bitkette s:

$$P_U(s) = \sum_{Prog.\ p:f_U(p)=s} 2^{-|p|}$$

Algorithmische Entropie der Bitkette s:

$$H_U(s) = -log_2 P_U(s).$$

Wegen Invarianztheorem:

$$H_{U_1}(s) = H_{U_2}(s) + O(1)$$

für 2 univ. TM U_1 and U_2. Daher

$$P_{U_1}(s) = P_{U_2}(s)2^{-O(1)} = P_{U_2}(s)O(1).$$

Daher ab jetzt: wir wählen eine universelle TM U und schreiben $P(s) = P_U(s)$. Rechtfertigung des Namens "universelle a priori Wahrsch.".

3.2 DOMINANZ DER KÜRZESTEN PROGRAMME

Es läßt sich zeigen:

$$K(s) = H(s) + O(1). \tag{3.1}$$

Da $H(s) = -logP(s)$, gilt

$$P(s) = (\frac{1}{2})^{K(s)-O(1)} = 2^{O(1)}2^{-K(s)} = O(2^{-K(s)}).$$

Die Wahrsch., irgendeins von s' Programmen zu raten, ist im wesentlichen gleich der Wahrsch., s' kürzestes Programm zu raten. Die Wahrsch. von s wird *dominiert* von seinen kürzesten Programmen.

Beweis z.B. [21] S. 224.

Kapitel 4

ALGORITHMISCHE INFORMATION

4.1 KONVENTIONELLE INFORMATIONSTHEORIE

Nach [37].

- Informationsmaß H: n Ereignisse mit Wahrsch. $p_1, p_2, ... p_n$. $\sum_i p_i = 1$. Entropie $H(p_1, p_2, \ldots, p_n)$ soll sein: Erwartungswert des Informationsgehaltes eines Zeichens.

$$H = -K \sum_i p_i log p_i, \quad (K \; const. \; > 0).$$

  Ist additiv: $H(X) + H(Y) = H(X, Y)$ falls Zv. X,Y unabh.

- bedingte Entropie:

$$H(Y \mid X) = -\sum_{i,j} p(i,j) log p(j \mid i)$$

- Wechselseitige Information zw. X und Y:

$$I(X, Y) = H(X) - H(X \mid Y) = H(X) + H(Y) - H(X, Y).$$

- Kullback-Leiber Distanz [15]: Zwei Verteilungen, T und P. α indiziert die mögl. Fälle.

$$G(T; P) = \sum_\alpha T_\alpha log \frac{T_\alpha}{P_\alpha}$$

  Interessant:

$$I(X, Y) = \sum_{x,y} p(x, y) log \frac{p(x, y)}{p(x)p(y)} = G(gemeinsame \; Vert.v.X, Y; unabh. \; Vert.v.X, Y)$$

4.2 ALGORITHMISCHE INFORMATIONSTHEORIE

Bedingte algorithmische Komplexität. Angenommen, TM U beginnt ihre Rechnung, wenn bereits eine nichtleere Bitkette x auf Rechenband steht. *Bedingte algorithmische Komplexität*

$$K(s \mid x) = min_p\{\mid p \mid: \; f_U(p) = s, \; bei \; gegebenem \; x \; auf \; Rechenband\}.$$

Sagt: Wieviel Information muss man zu dem addieren, was man schon kennt (x), um s zu erhalten.

Weiter definiert man *bedingte algorithmische Wahrscheinlichkeit*

$$P(s \mid x) = \frac{P(s,x)}{P(x)}$$

und *bedingte algorithmische Entropie*

$$H(s \mid x) = -logP(s \mid x).$$

Wir haben also

$$H(s,x) = H(s) + H(x \mid s).$$

4.2.1 GRUNDLEGENDE BEZIEHUNGEN

$$H(s,x) = H(x,s) + O(1).$$

$$H(s) \leq H(s,x) + O(1).$$

$$H(s,t) \leq H(s) + H(t) + O(1).$$

Wir wissen ja schon:

$$H(s) = K(s) + O(1).$$

Daraus ergeben sich viele Möglichkeiten, obige Beziehungen mit Hilfe von Kolmogorov Komplexität K umzuschreiben.

Anmerkung: Für fast alle s ist

$$K(s) =\mid s \mid + K(\mid s \mid).$$

Die Information, die das haltende Programm über die Länge seiner eigenen Ausgabe trägt, beträgt nämlich $K(\mid s \mid)$. $O(logs)$ ist obere Schranke für $K(\mid s \mid)$. Also:

$$K(s) <=\mid s \mid + O(logs) + O(loglogs) + O(logloglogs)...$$

Hieraus ergeben sich erneut viele Möglichkeiten, Beziehungen aufzustellen. Hausaufgabe!

Kapitel 5

MASCHINELLES LERNEN

Hierzu gibt's zusätzliches, separates, in der Vorlesung besprochenes, weiterführendes Material, insbesondere Hutter's Folien. Hier nur ein grober Überblick über ein paar relevante Konzepte.

5.1 INDUKTIVE INFERENZ, GENERALISIERUNGSFÄHIGKEIT

Ockhams Rasiermesser bevorzugt Problemlösungen, deren minimale Beschreibungslängen gering sind, gegenüber Problemlösungen, deren minimale Beschreibungslängen groß sind. Begründung durch algorithmische Wahrscheinlichkeitstheorie:

Das Problem bestehe darin, eine Symbolsequenz zu extrapolieren (o.B.d.A.: Bitsequenz). Wir haben den Bitstring s bereits beobachtet und wollen das nächste Bit vorhersagen. Das Ereignis "s wird vom Symbol i gefolgt" sei für $i \in \{0,1\}$ mit si bezeichnet. Mit Bayes:

$$P(s0 \mid s) = \frac{P(s \mid s0)P(s0)}{P(s)} = \frac{P(s0)}{P(s)}, \quad P(s1 \mid s) = \frac{P(s1)}{P(s)}.$$

Wir prophezeihen "das nächste Bit wird 0 sein", falls $P(s0) > P(s1)$, und umgekehrt. Da $P(si) = O((\frac{1}{2})^{K(si)})$ für $i \in \{0,1\}$, wird die Fortsetzung mit niedrigerer Kolmogorov Komplexität (i.a.) die wahscheinlichere sein.

5.2 MINIMALE BESCHREIBUNGSLÄNGE

Typisch: Gegeben Trainingsdaten D. Gesucht: die wahrscheinlichste Hypothese H, die D erklärt. Bayes liefert:

$$P(H \mid D) = \frac{P(D \mid H)P(H)}{P(D)}. \tag{5.1}$$

Wähle H so, daß $P(H \mid D)$ maximal. Äquivalenterweise: Minimiere

$$-logP(H \mid D) = -logP(D \mid H) - logP(H) + logP(D). \tag{5.2}$$

Interpretation: Da D gegeben ist, läßt sich $P(D)$ als normalisierende Konstante ignorieren. $-logP(D \mid H)$ ist als die Information interpretierbar, die benötigt wird, um D aus H zu berechnen. $-log\ P(H)$ ist die minimale Beschreibungslänge von H.

Ockhams Rasiermesser: Hat man univ. *a priori* Wahrsch. so daß $P(H)$ groß für einfache H: Ockham automatisch. Die einfachen (kurzen) Hypothesen sind wahrscheinlicher, also "besser". Ein guter Lernalgorithmus findet die kurzen!

Daraus ergibt sich das

Prinzip der minimalen Beschreibungslänge

("Minimal Description Length"(MDL)-Verfahren): **Minimiere Summe aus Information der Hypothese und Information der Rekonstruktionsfehler bei gegebener Hypothese!**

Da universelle Verteilung unberechenbar, nehmen viele statt der universellen Verteilung was anderes, z.B. Gaussverteilungen. Bei neuronalen Netzen z.B. Hinton [10]: Gaussrauschen auf den Gewichten, je mehr Rauschen, desto weniger Information. Wieder: Minimiere Information der Gewichte plus Information der Rekonstruktionsfehler, relativ zur Gaussverteilung.

Kapitel 6

OPTIMALE SUCHALGORITHMEN

Hierzu gibt es zusätzliches, separates, in der Vorlesung besprochenes, weiterführendes Material, insbesondere [11] (the "fastest" algorithm for all well-defined problems), und [35] (optimal oredered problem solver - incremental extension of universal search). Hier nur ein grober Überblick über ein paar relevante Konzepte.

6.1 LEVIN KOMPLEXITÄT

Erweiterung des Programmbegriffs: verschiedene Resultate, abhängig von der Laufzeit. Programm q ist Bitkette auf Eingabeband, die sich von U vollständig lesen läßt, bevor U Bitkette s auf das Ausgabeband geschrieben hat (U muß nicht halten). Sei $t(q, s)$ die Zahl der ausgeführten Rechenschritte, die notwendig sind, um s auszugeben. Dann Levin Komplexität:

$$Kt_U(s) = \min_q\{\mid q \mid + log\ t(q, s)\}.$$

Entsprechendes Invarianztheorem gilt auch für Kt. Also schreiben wir Kt statt Kt_U.

6.2 OPTIMALE UNIVERSELLE SUCHE

Levin Komplexität führt zu einem Suchalgorithmus, der Lösungen für viele wichtige Probleme in minimaler Zeit findet.

- Sei ϕ_i wieder die i-te Funktion in einer effektiven Aufzählung aller partiellen Funktionen von D_N auf D_N. Sei T_i die zu ϕ_i korrespondierende TM. Ist $\phi(y) = x$, dann heißt y ein ϕ-Zeuge von x.
- Inversionsprobleme: Algorithmus A invertiert Problem ϕ, falls A bei gegebenem x einen ϕ-Zeugen für x berechnet und nachweist, daß $\phi(y) = x$.
- Beispiele: Primzahlfaktorisierung, Auffinden einer Variablenbelegung, so daß gegebene Boolsche Formel erfüllt ist, etc.
- Naiver Ansatz: Erschöpfende Suche unter exponentiell vielen Lösungskandidaten (Zeugenkandidaten).

OPTIMALE ALTERNATIVE

- Statt dessen: Alg. SIMPLE: Laß T_1 jeden zweiten Zeitschritt einen Schritt ausführen. Laß T_2 jeden zweiten der verbleibenden Zeitschritte einen Schritt ausführen. Laß T_3 jeden zweiten der verbleibenden Zeitschritte einen Schritt ausführen. Usw...

  Falls T_k ϕ in t Schritten invertiert, invertiert SIMPLE ϕ in $2^k t + 2^{k-1}$ Schritten.

  Gibt es also einen Alg., der ϕ in $t(n)$ Schritten invertiert (n ist Problemgröße), dann invertiert SIMPLE ϕ in $ct(n)$ Schritten, wobei $c = 2^{k+1}$ konstant und unabhängig vom Problem ist.

- **Theorem** (Verbesserung der Konstante): Gibt es einen Alg. T, der ϕ in $t(n)$ Schritten invertiert (n ist Problemgröße), dann invertiert UNIVERSELLE SUCHE (siehe unten) ϕ in $ct(n)$ Schritten, wobei $c = 2^{K(T)+1}$.

  Definiere hierzu erst

  $$Kt'(w \mid x, \phi) = \min_q \{\mid p \mid + log\, t(p,w) : in\, t(p,x)\ Schritten\ druckt\ p\ w\ und\ testet,\ ob\ \phi(w) = x\ bei\ geg.\ x\}.$$

  UNIVERSELLE SUCHE generiert bei gegebenen (x, ϕ) alle Zeichenketten w geordnet nach wachsendem $Kt'(w \mid x, \phi)$ und prüft, ob $\phi(w) = x$, bis ein ϕ-Zeuge für x gefunden ist. Das geht so:

  ALGORITHMUS: Alle selbstbeschränkenden Programme p mit $\mid p \mid < i$ werden in Phase i für $2^i 2^{-|p|}$ Schritte ausgeführt, bis ϕ bei x invertiert wurde.

  Dabei werden alle Strings w mit $Kt'(w \mid x, \phi) \leq k$ in 2^{k+1} Zeitschritten generiert und getestet.

  Beweis: Falls $Kt'(w \mid x, \phi) \leq i$, dann $\mid p \mid + log\ t(p,x) \leq i$. Also $t(p,x) \leq 2^{i-|p|}$, der Zeit, die für p in der i-ten Phase zur Verfügung gestellt wurde. Wir haben nämlich

  $$P(s) = \sum_{Prog.\ p\ hält: f(p)=s} 2^{-|p|} \leq 1.$$

  Also

  $$\sum_{1 \leq i \leq k} \sum_{0 \leq i-|p|} 2^{i-|p|} \leq \sum_{p\ hält} 2^{-|p|} \sum_{1 \leq i \leq k} 2^i \leq 2^{k+1}$$

  Sei $m = min\{Kt'(w \mid x, \phi)$: w ist ϕ-Zeuge für $x\}$. Sei $n :=\mid x \mid$. Angenommen, es gibt TM T, die ϕ bei x in $t(n)$ Schritten invertiert. Nach Definition: $m \leq Kt'(T \mid x, \phi)$. UNIVERSELLE SUCHE invertiert ϕ in c^{m+1} Schritten (siehe oben). Nach Definition: $Kt'(T \mid x, \phi) \leq K(T) + logt(n)$. Also braucht UNIVERSELLE SUCHE höchstens $2^{K(T)+1} t(n)$ Schritte, was zu zeigen war.

  Ist der Inversionsalgorithmus T_k für ein bestimmtes Problem also selbst besonders einfach, kann im Vergleich zu SIMPLE noch viel gewonnen werden.

PROBABILISTISCHE VARIANTE

- *Zeitlimits würfeln: Wahrscheinlichkeit einer bestimmten Zahl von Zeitschritten ist dabei proportional zum Logarithmus der Zahl. wie in Abschnitt 4.1 Programme würfeln und ausführen, bis ϕ bei x invertiert wurde.*

- Anwendung für konventionelle digitale Maschine: wird in Vorlesung besprochen.

Kapitel 9

HISTORISCHES

History spotlights. In 1965, A. N. Kolmogorov (1903-1987), founder of modern axiomatic probability theory [13], was the first to introduce a variant of the complexity measure K for its own sake [14]. Levin (1984) cites announcements of Kolmogorov's lectures on this subject dating back to 1961. In independent and even earlier work, R. J. Solomonoff (1964) had already come up with the same measure as a by-product of his work on algorithmic probability and inductive inference (a preliminary version of his paper is dated 1960). Both Solomonoff and Kolmogorov observed K's machine independence. Today, even Solomonoff himself refers to K as "Kolmogorov complexity", e.g. [39]. In 1969, G. J. Chaitin independently also published the essential concepts [3] (some hints were already provided at the end of his 1966 paper). Important related early work is described in [23, 7, 36]. Apparently, L. A. Levin was the first to introduce and analyze today's "standard form" of Kolmogorov complexity based on halting programs and prefix codes [18], see also [7, 16, 19, 43]. Levin proved $K(s) = H(s) + O(1)$. The importance of prefix codes was independently seen by Chaitin (1975), who also proved the above equation and attributes part of the argument to N. Pippenger. Levin introduced Kt complexity and the universal optimal search algorithm (see e.g. [17] and [20], where related ideas are attributed to Adleman). Other generalizations of Kolmogorov complexity have been proposed, e.g. [9], but see the contributions in [42] for more. Easily computable approximations of the MDL principle were formulated by Wallace and Boulton (1968) and Rissanen (1978, 1983, 1986). Such approximations build the basis of most if not all current machine learning applications, e.g. [27, 8, 24, 26]. Barzdin, referred to in [43], related Kolmogorov complexity to a variant of Gödel's incompleteness theorem, a subject which became a central theme of Chaitin's research [5]. Meanwhile, the theory of Kolmogorov complexity has split into many subfields. An excellent overview and many additional details on the history are given in Li and Vitanyi's book (1993). See also [6]. See [32] for the first (?) machine learning application of universal search. See [33] for the first application to fine arts. See [11] for the "fastest" algorithm for all well-defined problems. See [35] for the incremental extension of universal search. See [12] for a theory of (incomputable) universal intelligence based on Solomonoff's prior. This lecture is partly inspired by presentations found in [5], [22], [39], [12].

Literaturverzeichnis

[1] Y. M. Barzdin. Algorithmic information theory. In *Encyclopaedia of Mathematics*, volume 1, pages 140–142. Reidel, Kluwer Academic Publishers, 1988.

[2] G. J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the ACM*, 13:547–569, 1966.

[3] G. J. Chaitin. On the length of programs for computing finite binary sequences: statistical considerations. *Journal of the ACM*, 16:145–159, 1969.

[4] G. J. Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM*, 22:329–340, 1975.

[5] G. J. Chaitin. *Algorithmic Information Theory.* Cambridge University Press, Cambridge, 1987.

[6] T. M. Cover, P. Gács, and R. M. Gray. Kolmogorov's contributions to information theory and algorithmic complexity. *Annals of Probability Theory*, 17:840–865, 1989.

[7] P. Gács. On the symmetry of algorithmic information. *Soviet Math. Dokl.*, 15:1477–1480, 1974.

[8] Q. Gao and M. Li. The minimum description length principle and its application to online learning of handprinted characters. In *Proc. 11th IEEE International Joint Conference on Artificial Intelligence, Detroit, Mi*, pages 843–848, 1989.

[9] J. Hartmanis. Generalized Kolmogorov complexity and the structure of feasible computations. In *Proc. 24th IEEE Symposium on Foundations of Computer Science*, pages 439–445, 1983.

[10] G. E. Hinton and D. van Camp. Keeping neural networks simple. In *Proceedings of the International Conference on Artificial Neural Networks, Amsterdam*, pages 11–18. Springer, 1993.

[11] M. Hutter. The fastest and shortest algorithm for all well-defined problems. *International Journal of Foundations of Computer Science*, 13(3):431–443, 2002. (On J. Schmidhuber's SNF grant 20-61847).

[12] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability.* Springer, Berlin, 2004. (On J. Schmidhuber's SNF grant 20-61847).

[13] A. N. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung.* Springer, Berlin, 1933.

[14] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:1–11, 1965.

[15] S. Kullback. *Statistics and Information Theory.* J. Wiley and Sons, New York, 1959.

[16] L. A. Levin. On the notion of a random sequence. *Soviet Math. Dokl.*, 14(5):1413–1416, 1973.

[17] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

[18] L. A. Levin. Laws of information (nongrowth) and aspects of the foundation of probability theory. *Problems of Information Transmission*, 10(3):206–210, 1974.

[19] L. A. Levin. Various measures of complexity for finite objects (axiomatic description). *Soviet Math. Dokl.*, 17(2):522–526, 1976.

[20] L. A. Levin. Randomness conservation inequalities: Information and independence in mathematical theories. *Information and Control*, 61:15–37, 1984.

[21] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, 1993.

[22] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications (2nd edition)*. Springer, 1997.

[23] P. Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.

[24] A. Milosavljevič and J. Jurka. Discovery by minimal length encoding: A case study in molecular evolution. *Machine Learning*, 12:96–87, 1993.

[25] F. Nake. *Ästhetik als Informationsverarbeitung*. Springer, 1974.

[26] E. P. D. Pednault. Some experiments in applying inductive inference principles to surface reconstruction. In *11th IJCAI*, pages 1603–1609. Morgan Kaufmann, 1989.

[27] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80:227–248, 1989.

[28] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[29] J. Rissanen. A universal prior for integers and estimation by minimum description length. *The Annals of Statistics*, 11(2):416–431, 1983.

[30] J. Rissanen. Stochastic complexity and modeling. *The Annals of Statistics*, 14(3):1080–1100, 1986.

[31] J. Schmidhuber. Low-complexity art. Technical Report FKI-197-94, Fakultät für Informatik, Technische Universität München, 1994.

[32] J. Schmidhuber. Discovering solutions with low Kolmogorov complexity and high generalization capability. In A. Prieditis and S. Russell, editors, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 488–496. Morgan Kaufmann Publishers, San Francisco, CA, 1995.

[33] J. Schmidhuber. Low-complexity art. *Leonardo, Journal of the International Society for the Arts, Sciences, and Technology*, 30(2):97–103, 1997.

[34] J. Schmidhuber. Gödel machines: self-referential universal problem solvers making provably optimal self-improvements. Technical Report IDSIA-19-03, arXiv:cs.LO/0309048, IDSIA, Manno-Lugano, Switzerland, 2003.

[35] J. Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54:211–254, 2004.

[36] C. P. Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory*, 5:246–258, 1971.

[37] C. E. Shannon. A mathematical theory of communication (parts I and II). *Bell System Technical Journal*, XXVII:379–423, 1948.

[38] R. J. Solomonoff. A formal theory of inductive inference. Part I. *Information and Control*, 7:1–22, 1964.

[39] R. J. Solomonoff. An application of algorithmic probability to problems in artificial intelligence. In L. N. Kanal and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 473–491. Elsevier Science Publishers, 1986.

[40] R. von Mises. *Probability, Statistics, and Truth*. MacMillan, 1939.

[41] C. S. Wallace and D. M. Boulton. An information theoretic measure for classification. *Computer Journal*, 11(2):185–194, 1968.

[42] O. Watanabe. *Kolmogorov complexity and computational complexity*. EATCS Monographs on Theoretical Computer Science, Springer, 1992.

[43] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the algorithmic concepts of information and randomness. *Russian Math. Surveys*, 25(6):83–124, 1970.