Kolmogorow-Komplexität

Tobias Schomann, AuD2, Institut für Theoretische Informatik, Universität zu Lübeck

Beispiel

Seien zwei Zeichenketten gegeben:

(1) 1101001001
(2) 0101010101

Unter Verwendung der Informationstheorie von Shannon messen wir den erwarteten Informationsgehalt, der sich auf einen Übertragungskanal und somit auf eine gemeinsame Gesamtmenge möglicher Zeichenketten bezieht. Ausschlaggebend ist dabei die Auftrittswahrscheinlichkeit der Symbole. Demnach haben die obigen Zeichenketten den gleichen Informationsgehalt, da sie aus gleich vielen Nullen und Einsen bestehen.

Bei der Kolmogorow-Komplexität betrachten wir jede Zeichenkette für sich und stellen die Frage, ob es einen einfachen Algorithmus gibt, der das Wort konstruieren kann. Demnach würde man zunächst Annehmen, dass das zweite Wort weniger Information enthält, denn es lässt sich durch den einfachen Ausdruck $(01)^5$ definieren. Die erste Zeichenkette wirkt dagegen eher zufällig.

Definition

Es sei ein Alphabet Σ, dass $\square$ nicht enthält, und eine universelle Turingmaschine U mit zwei Bändern über $\Sigma \cup \{\square\}$. Wenn in der Startkonfiguration von U eine Zeichenkette $p \in \Sigma^*$ auf dem zweiten Band steht, während das erste Band leer bleibt, und U in endlicher Zeit einen akzeptierenden Zustand mit einem String $x \in \Sigma^*$ auf dem ersten Band erreicht, dann bezeichnen wir p als Programm, dass x auf U ausgibt.

Die Kolmogorow-Komplexität eines Strings $x \in \Sigma^*$ ist nun definiert durch

$$K_U(x) = \min\{|p| : p \text{ gibt } x \text{ auf } U \text{ aus}\}$$

Eigenschaften von K

Eine ungünstige Konstruktion von U kann problematisch sein. Wird zum Beispiel nur jedes zweite Symbol auf dem Eingabeband beachtet, dann ist die Komplexität doppelt so groß. Daher wird o.B.d.A. folgende Annhame getroffen.

$\exists d \in \Sigma^*$:

(1) Jede Turingmaschine kann durch ein Programm simuliert werden, das d nicht enthält.

(2) Wird U mit dem Programm xdy auf dem zweiten Band gestartet, verhält sie sich, als würde sie mit x auf dem zweiten und y auf dem ersten Band gestartet.

Lemma Es existiert eine Konstante c_U, s. d. $K_U(x) \leq |x| + c_U$

Beweis: Sei $x \in \Sigma^*$ und p_0 das Programm der trivialen TM, die sofort anhält. Dann folgt aus der Annahme (2), dass das Programm $p_0 dx$ auf U nach endlicher Zeit anhält und x ausgibt. Also gilt $K_U(x) \leq |p_0| + |d| + |x|$, wobei p_0 und d nur von U abhängen. $\square$

Invarianz-Theorem Sei $x \in \Sigma^*$ und seien S, T universelle Turingmaschinen. Dann existiert eine Konstante c_{ST}, s. d. $|K_S(x) - K_T(x)| \leq c_{ST}$.

Beweis: Sei ein Programm p, das x auf S ausgibt und eine einbändige Turingmaschine S_0, die S so simuliert, dass in ihrer Startkonfiguration p auf dem Band steht und in einer akzeptierenden Konfiguration ausschließlich

x auf dem Band enthalten ist. Nun sei p_{S_0} das Programm für T, das S_0 simuliert, und p_x das kürzeste Programm, das x auf S ausgibt. Das Programm $p_{S_0}dp_x$ gibt dann x auf T aus. Somit gilt $K_S(x) = |p_x|$ und $K_T(x) \leq |p_{S_0}| + |d| + |p_x|$, wobei d konstant ist und p_{S_0} hängt von S und T ab und ist daher ebenfalls konstant. □

Aufgrund dieses Resultats kann bei asymptotischer Betrachtung beliebiges festes Maschinenmodell angenommen werden. Man lässt dann den Index weg und schreibt $K(x)$.

Theorem K ist nicht berechenbar.

Beweis: Der Beweis erfolgt indirekt. Wir nehmen an K sei berechenbar. Weiterhin sei $c \in \mathbb{N}$, der Raum Σ^* lexikografisch geordnet und die Funktion $x(k)$ liefere das k-te Element aus Σ^*. x_0 sei definiert als das erste Wort in Σ^*, für das gilt $K(x_0) \geq c$. Da $K(x)$ und offensichtlich auch $x(k)$ berechenbar sind, können wir zur Berechnung von x_0 folgendes Programm konstruieren.

1. $k \leftarrow 0$
2. **while** $K(x(k)) < c$ **do** $k \leftarrow k + 1$
3. **return** $x(k)$

Abgesehen von c, welches Teil des Programms ist, hat das Programm eine konstante Größe und c lässt mit logarithmischen Platzbedarf codieren. Es gilt also $K(x_0) \leq \log c + O(1)$, was für ausreichend große c ein Widerspruch ist. Folglich ist K nicht berechenbar. □

Theorem K ist von oben rekursiv aufzählbar. (Oder anders gesagt: K ist von oben approximierbar.)

Da eine obere Schranke für die Länge des kürzesten Programm bekannt ist, gibt es endlich viele Programme die in Frage kommen. Sie können mit einer Zeitschranke der Reihe nach simuliert werden. Durch das schrittweise Erhöhen der Zeitschranke können dann immer kürzere Programme entdeckt werden, sofern diese existieren.

Definition Sei $x \in \Sigma^*$, dann heißt x komprimierbar, gdw. $K(x) < |x|$.

Theorem Es existieren $x \in \Sigma^*$, die nicht komprimierbar sind.

Beweis: Sei $m = |\Sigma|$ und $n = |x|$. Die Anzahl der Programme, die kleiner als n sind, ist $m^0 + m^1 + ... + m^{n-1}$. Die Anzahl der Zeichenketten $x \in \Sigma^*$ der Länge n ist m^n. Es gilt

$$m^0 + m^1 + ... + m^{n-1} < m^n \quad .$$

Folglich existiert ein x der Länge n mit $K(x) \geq n$. □

Einfache Anwendungsbeispiele

- $K(0^n) \leq \log n + O(1)$
- $K(\pi_n) \leq \log n + O(1)$
- Palindrom p_n, $K(p_n) \leq \lceil n/2 \rceil + O(1)$

Literatur

[1] László Lovász. *Information complexity: the complexity-theoretic notion of randomness* in *Computation complexity*. Lecture notes, 1994.

[2] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, New York, 1997.