Algorithmen und Wahrscheinlichkeit

Woche 9

Ilya Maier

Abschätzen von Wahrscheinlichkeiten

Markovs Ungleichung:

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t} \qquad \forall X \geq 0 \quad \forall t > 0,\ t \in \mathbb{R}$$

Chebyshevs Ungleichung:

$$\Pr[\,|X - \mathbb{E}[X]\,| \geq t] \leq \frac{\mathrm{Var}[X]}{t^2} \qquad \forall X \quad \forall t > 0,\ t \in \mathbb{R}$$

Ist nun Markov oder Chebyshev besser?

$$\Pr[X \geq t] \quad \leq \quad \Pr[X - \mathbb{E}[X] \geq t - \mathbb{E}[X]] \quad \leq \quad \Pr[\,|X - \mathbb{E}[X]\,| \geq t - \mathbb{E}[X]] \quad \leq \quad \frac{\mathrm{Var}[X]}{(t - \mathbb{E}[X])^2}$$

d.h. $t > \mathbb{E}[X]$, also WLOG sei $t = \mathbb{E}[X] + k \cdot \sqrt{\mathrm{Var}[X]}$ für $k > 0$

dann $\dfrac{\mathrm{Var}[X]}{(t - \mathbb{E}[X])^2} = \dfrac{\mathrm{Var}[X]}{k^2 \mathrm{Var}[X]} = \dfrac{1}{k^2}$, wenn also $k^2 \mathbb{E}[X] > \mathbb{E}[X] + k\sqrt{\mathrm{Var}[X]}$ ist, dann ist Chebyshev besser

Abschätzen von Wahrscheinlichkeiten

Chernoffs Ungleichung:

1. $\Pr[X \geq (1+\delta)\mathbb{E}[X]] \leq e^{-\frac{1}{3}\delta^2\mathbb{E}[X]}$ $\quad \forall X \sim \text{Bin}(n,p)$
2. $\Pr[X \leq (1-\delta)\mathbb{E}[X]] \leq e^{-\frac{1}{2}\delta^2\mathbb{E}[X]}$ $\quad \forall 0 < \delta \leq 1$

Korollar:

1. $\Pr[X - \mathbb{E}[X] \geq \delta\mathbb{E}[X]] \leq e^{-\frac{1}{3}\delta^2\mathbb{E}[X]}$
2. $\Pr[X - \mathbb{E}[X] \leq -\delta\mathbb{E}[X]] \leq e^{-\frac{1}{2}\delta^2\mathbb{E}[X]}$
3. $\Pr[\,|X - \mathbb{E}[X]| \geq \delta\mathbb{E}[X]] \leq e^{-\frac{1}{3}\delta^2\mathbb{E}[X]} + e^{-\frac{1}{2}\delta^2\mathbb{E}[X]} \leq 2e^{-\frac{1}{3}\delta^2\mathbb{E}[X]}$

Target Shooting

Gegeben: endliche Mengen S, U s.d. $S \subseteq U$,

$$I_S : U \to \{0,1\} : I_S(u) = 1 \iff u \in S$$

Gesucht: $|S|/|U|$

Algorithmus

1) wähle $u_1, \ldots, u_N$ aus U zufällig unabhängig und gleichverteilt

2) **return** $Y := \frac{1}{N} \sum_{i=1}^{N} I_S(u_i) \qquad \Longrightarrow \qquad \mathbb{E}[Y] = \frac{|S|}{|U|}$

Satz: (geeignetes N finden)

Für $\delta, \epsilon > 0$:

$$N \geq 3 \frac{|U|}{|S|} \epsilon^{-2} \ln\left(\frac{2}{\delta}\right) \qquad \Longrightarrow \qquad \Pr\left[\left|Y - \mathbb{E}[Y]\right| \geq \epsilon \cdot \mathbb{E}[Y]\right] \leq \delta$$

(Kann man mit Chernoffs Ungleichungen i) und ii) zeigen)

Primzahltest

A. GGT

$\gcd(a, n) > 1$ für $1 \leq a \leq n-1$

$\Longrightarrow$ n nicht prim

i) “nicht prim” immer richtig

ii) $\Pr[\text{"prim"} \mid \text{nicht prim}] = \dfrac{|Z_n^*|}{n-1}$

iii) $\text{cost}(\gcd(m, n)) = \mathcal{O}((\log nm)^3)$

B. Fermat’s little theorem

n ist prim

$\Longrightarrow \forall a \in [n-1] : a^{n-1} \equiv 1 \mod n$

i) “nicht prim” immer richtig

ii) $\Pr[\text{"prim"} \mid \text{nicht prim}] = \dfrac{|PB_n|}{n-1}$

iii) $PB_n := \{a \in \mathbb{Z}_n^* \mid a^{n-1} \equiv 1 \bmod n\}$

iv) Carmichael-Zahl n:

1) n ist nicht prim

2) $PB_n = \mathbb{Z}_n^*$

v) $\Pr[\text{"prim"} \mid \text{nicht prim}] < 0.5$,

falls n nicht Carmichael

$\Longrightarrow$ Verbesserung durch Wiederholung

C. Miller-Rabin

Miller-Rabin-PrimeTest(n)

1) Wähle $1 \leq a \leq n-1$ gleichverteilt zufällig
2) $d, k \in \mathbb{N}$ s.d. $n - 1 = 2^k d$, wobei d ungerade
3) **if** $a^d \mod n \neq 1$ **&&**
   $\nexists i < k : a^{2^i d} \mod n = n - 1$ **then**
   return “nicht prim"
4) **else return** "prim"

i) “nicht prim” immer richtig

ii) $\Pr[\text{"prim"} \mid \text{nicht prim}] \leq \dfrac{1}{4}$

$\Longrightarrow$ Verbesserung durch Wiederholung

iii) Laufzeit: $\mathcal{O}(\text{poly}(\log n))$

Duplikate finden

Datenmenge: $S = (s_1, \ldots, s_n)$, ggf. so, dass jeder s_i sehr viel Speicherplatz braucht (viel mehr als Integer)
Duplikate: (i, j) wobei $1 \leq i < j \leq n,$ falls $s_i = s_j$

A. Sortieren

1) Sortiere $((s_i, i))_{1 \leq i \leq n}$ nach s_i
2) Iteriere und finde Duplikate

i) Sortieren: $\mathcal{O}(n \log n)$
ii) Iterieren: $\mathcal{O}(n + |\,\text{Dupl}(S)\,|)$

B. Hashing

1) Hashe S mit $h : S \to [m]$

$$\Pr[h(s) = i] = \frac{1}{m}$$

2) Sortieren
3) Iterieren

i) Hashen: $\mathcal{O}(n)$
ii) Sortieren: $\mathcal{O}(n \log n)$
iii) Iterieren: $\mathcal{O}(n + |\,\text{Dupl}(S)\,| + \#\text{Kollisionen})$

Kollisionen: (i, j), wobei $s_i \neq s_j \wedge h(s_i) = h(s_j)$

$$\mathbb{E}[\#\text{Kollisionen}] \leq \binom{n}{2}\frac{1}{m}$$

C. Bloom Filter

1) Hashe S mit k Hashfunktionen $h_i : S \to [m]$
2) Wenn alle $h_i(s_j)$ Werte vorgekommen sind fügen wir s_j in L hinzu.

i) Hashen: $\mathcal{O}(kn)$
ii) Iteration: $\mathcal{O}(n + n \cdot |\,L\,|)$

$$\mathbb{E}[\#\text{falsche } L\text{-Einträge}] \leq n\left(1 - \left(1 - \frac{1}{m}\right)^{k(n-1)}\right)^k$$

Kahoot

Aufgaben