Karlsruher Institut für Technologie

Maximilian Katzmann
Hans-Peter Lehmann
Stefan Walzer

Übungsblatt 04

Randomisierte Algorithmik – Wintersemester 2023/2024

Aufgaben zur Vorlesung vom 16.11.2023
Abgabe im ILIAS bis 23.11.2023, 11:30 Uhr
Besprechung am 05.12.2023, 08:00 Uhr

Achte insbesondere bei handschriftlichen Abgaben auf Lesbarkeit. Die Abgabe erfolgt über das Übungsmodul im ILIAS. Gib Deine Ausarbeitungen in *einer* PDF-Datei ab.

Aufgabe 1 – Obere Schranken für obere Schranken

In der Vorlesung haben wir Chernoff Schranken kennengelernt, die insbesondere bei Summen Bernoulli-verteilter Zufallsvariablen nützlich sind (siehe die Korollare am Ende des Foliensatzes).

Für Bernoulli-Zufallsvariablen $X_1, ..., X_n$ und deren Summe $X = \sum_{i \in [n]} X_i$ kann für ein $\varepsilon \in (0, 1]$ damit $\Pr[X \geq (1 + \varepsilon)\mathbb{E}[X]]$ nach oben beschränkt werden. Intuitiv sollte die Wahrscheinlichkeit, dass X noch größer ist, umso kleiner sein.

a) Zeige, dass für eine obere Schranke $f(n) \geq \mathbb{E}[X]$ gilt, dass $\Pr[X \geq (1 + \varepsilon)f(n)] \leq e^{-\varepsilon^2/3 \cdot f(n)}$.

b) Zeige, dass für $f(n) \geq \mathbb{E}[X]$ mit $f(n) \in \Omega(\log(n))$ gilt, dass $X \in O(f(n))$ mit Wahrscheinlichkeit $1 - O(n^{-c})$ für eine beliebige Konstante $c > 0$.

Aufgabe 2 – Sammlerstück

Wir tauchen ein in die Welt der Parallelisierung. Dazu betrachten wir m Prozessoren, auf denen Jobs verteilt werden können. Normalerweise wird jeder Job in einem einzigen Zeitschritt ausgeführt. Ab und zu kann es jedoch passieren, dass der Garbage-Collector angeworfen werden muss, was die Ausführungszeit eines Jobs auf $k > 1$ erhöht. Wenn man insbesondere annimmt, dass n Jobs auf den m Prozessoren verteilt werden, wird jeder Job unabhängig mit Wahrscheinlichkeit $p = m \log(nm)/(6n)$ ausgebremst.

Wir gehen nun davon aus, dass die n Jobs so verteilt wurden, dass jeder Prozessor genau n/m Jobs nacheinander bearbeitet. Zeige, dass mit Wahrscheinlichkeit $1 - 1/n$ alle Jobs nach $n/m + (k - 1)\log(nm)$ Schritten abgeschlossen sind.
Hinweis: Wir gehen davon aus, dass n ein Vielfaches von m ist.

Aufgabe 3 – Bonus – Geometric Jumps

Wie in der Vorlesung erwähnt ist ein nützliches Kriterium eines Graphenmodells, dass man daraus effizient Graphen ziehen kann. Im Folgenden wollen wir uns anschauen, wie das für Erdős-Rényi Graphen $G(n,p)$ geht.

Ein trivialer Algorithmus sampelt dafür $m = \binom{n}{2}$ unabhängige Indikatorzufallsvariablen $X_1, ..., X_m \sim \mathrm{Ber}(p)$, eine für jede Kante. Wir wollen das effizienter machen, zumindest erwartet. Um den Graph zu erstellen, müssen wir nur wissen, welche Kanten es geben soll, also welche $X_i = 1$ sind. Demnach würden wir die X_i mit $X_i = 0$ gern überspringen.

Formal wollen wir aus einer Menge $M = [m]$ eine Teilmenge X auswählen, die jedes Element unabhängig mit Wahrscheinlichkeit p enthält. Dazu nehmen wir uns eine Menge $Y \subseteq \mathbb{N}$ die *jedes* $i \in \mathbb{N}$ unabhängig mit Wahrscheinlichkeit p enthält und erhalten $X = Y \cap M$ als die gewünschte Menge.

a) Zeige, dass $\min(Y)$ eine Zufallsvariable ist, die der geometrischen Verteilung mit Parameter p folgt, also $\min(Y) \sim \mathrm{Geo}(p)$.
Hinweis: $\mathrm{Geo}(p)$ ist eine Verteilung auf $\mathbb{N} = \{1, 2, ...\}$ wobei für $Z \sim \mathrm{Geo}(p)$ gilt, dass $\Pr[Z = i] = p(1-p)^{i-1}$.

b) Zeige, dass für $Z_1, Z_2, ... \sim \mathrm{Geo}(p)$ gilt, dass die Menge der Präfixsummen $\{Z_1, Z_1 + Z_2, Z_1 + Z_2 + Z_3, ...\}$ die gleiche Verteilung hat wie Y.

Daraus folgt, dass man die Elemente von Y in aufsteigender Reihenfolge sampeln kann, wobei man in Schritt i das i-te Element bekommt.

Für $X = Y \cap M$ werden dafür dann $O(|X|)$ Schritte benötigt.

c) Zeige, dass man erwartet nur $O(pn^2 + n)$ Schritte braucht um $G(n,p)$ zu samplen.