Maximilian Katzmann
Hans-Peter Lehmann
Stefan Walzer

Übungsblatt 10

Randomisierte Algorithmik – Wintersemester 2023/2024

Aufgaben zur Vorlesung vom 11.1.2024
Abgabe im ILIAS bis 18.1.2024, 11:30 Uhr
Besprechung am 30.01.2024, 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.

Diesmal sind es zwar drei Aufgaben, aber sie sind nicht so schwierig. Versprochen!

Aufgabe 1 – Rechenregeln für Varianz

Seien X, Y unabhängige Zufallsvariablen deren Varianz existiert. Seien ferner $s, t > 0$. Zeige, dass gilt:

(a) $\mathrm{Var}(sX) = s^2\mathrm{Var}(X)$

(b) $\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y)$

(c) $\mathrm{Var}(sX + tY) = s^2\mathrm{Var}(X) + t^2\mathrm{Var}(Y)$

Beweiszutaten: Verwende die Definition $\mathrm{Var}(Z) := \mathbb{E}[(Z-\mathbb{E}[Z])^2]$. Verwende ferner, dass für unabhängige Zufallsvariablen X, Y gilt $\mathbb{E}[X \cdot Y] = \mathbb{E}[X] \cdot \mathbb{E}[Y]$. Wenn du Lust hast, kannst du letzteres auch nochmal aus der Definition von Unabhängigkeit (für diskrete Zufallsvariablen) herleiten, die lediglich $\Pr[X = i \wedge Y = j] = \Pr[X = i] \cdot \Pr[Y = j]$ für alle i, j garantiert.

Aufgabe 2 – Die Jensensche Ungleichung

Sei $D \subseteq \mathbb{R}$ ein zusammenhängender Definitionsbereich und $f : D \to \mathbb{R}$ eine Funktion. Die Funktion f heißt konvex, wenn sie „linksgekrümmt“ ist und konkav wenn sie „rechtsgekrümmt“ ist.[1] Eine Funktion ist konvex genau dann wenn ihre Negation konkav ist. Für eine formale Definition siehe: Wikipedia

(a) Entscheide (ohne Beweis) für folgende Funktionen, ob diese auf ihrem jeweiligen Definitionsbereich konvex ist, konkav ist, beides ist oder weder noch.

$$f_1(x) = x, \qquad f_2(x) = x^2, \qquad f_3(x) = x^3, \qquad f_4(x) = \log(x), \qquad f_5(x) = \log^2(x).$$

[1] Das „linksgekrümmt“ in Anführungszeichen lässt neben Linkskrümmungen (einer zweimal stetig differenzierbaren Funktion) auch Linksknicke und geradlinige Verläufe zu.

(b) Sei f eine konvexe Funktion mit Definitionsbereich D. Argumentiere geometrisch, dass es für jedes $x_0 \in D$ eine lineare Funktion g gibt, sodass gilt:

   (i) $f(x) \geq g(x)$ für alle $x \in D$

   (ii) $f(x_0) = g(x_0)$.

(c) Schlussfolgere, dass für jede konvexe Funktion f und für jede Zufallsvariable X mit Werten im Definitionsbereich D von f gilt:

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X]).$$

Hinweis: Betrachte $x_0 = \mathbb{E}[X]$ und das zugehörige g aus der vorherigen Teilaufgabe.

(d) Zeige, dass analog für jede konkave Funktion f mit Definitionsbereich D und für jede Zufallsvariable X mit Werten in D gilt:

$$\mathbb{E}[f(X)] \leq f(\mathbb{E}[X]).$$

Die Ungleichung aus (c) sowie Varianten wie in (d) nennt man Jensensche Ungleichung.

Aufgabe 3 – Analyse von Lossy Counting

Erinnerung: Lossy Counting ist ein einfacher Streaming Algorithmus, der die Länge m eines Streams approximativ zählt. Darin kommt ein Parameter $p \in (0, 1]$ vor. Der Algorithmus selbst sowie die Art und Weise, wie er verwendet wird, ist rechts nochmal zu sehen. Beweise:

(a) $\mathbb{E}[\text{result}] = m$

(b) $\Pr[|\text{result} - m| \leq \varepsilon m] \geq 1 - 2\exp(-\varepsilon^2 pm/3)$.
Hinweis: Nutze die Chernoff-Schranke die in der Vorlesung (sowohl im Approximationsteil als auch im Streamingteil vorgekommen ist. Diese ergibt sich übrigens direkt aus zwei Korollaren von der Folie „Chernoff – Simpler Versions“ des Concentration-Foliensatzes.

(c) $\mathbb{E}[\text{space}] \leq \log(1 + mp) + 1$.
Hinweis: Mit space bezeichnen wir den maximalen Speicherverbrauch, der für den Zustand Z von LossyCounting benötigt wird. Eine Zahl $i \in \mathbb{N}$ lässt sich mit $\lceil \log_2(i+1) \rceil$ Bits kodieren. Verwende die Jensensche Ungleichung aus Aufgabe 2.

Algorithm init:
- $Z \leftarrow 0$
- **return** Z

Algorithm update(Z, a)**:**
- **with probability** p **do**
  - $Z \leftarrow Z + 1$
- **return** Z

Algorithm result(Z)**:**
- **return** Z/p

Verwendung:
- $Z \leftarrow \text{init}()$
- **for** $i = 1$ **to** m **do**
  - $Z \leftarrow \text{update}(Z, a_i)$
- **return** result(Z)