Karlsruher Institut für Technologie

Stefan Walzer
ITI Prof. Sanders
Wintersemester 2023/2024

Übungsblatt 10 – Approximation Algorithms

Randomisierte Algorithmik

Aufgabe 1 – 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.

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

(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.

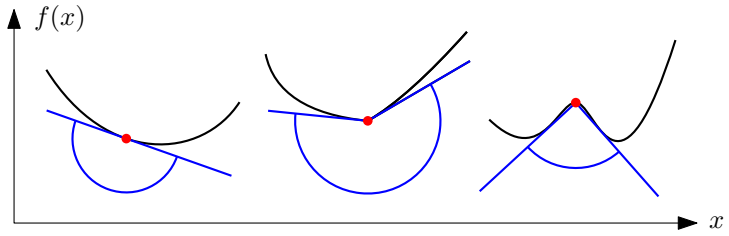
[1] Das „linksgekrümmt“ in Anführungszeichen lässt neben Linkskrümmungen (einer zweimal stetig differenzierbaren Funktion) auch Linksknicke und geradlinige Verläufe zu.

Lösung 1

(a) Die Funktionen sind alle zweimal stetig differenzierbar. Wenn die zweite Ableitung überall nicht-negativ ist, dann ist die Funktion konvex, wenn sie überall nicht-positiv ist, dann ist die Funktion konkav.

- f_1 ist konvex und konkav
- f_2 ist konvex
- f_3 ist weder konvex noch konkav
- f_4 ist konkav
- f_5 ist weder konvex noch konkav

(b) Weil f linksgekrümmt ist, kann man am Funktionsgraphen von f an Stelle $(x_0, f(x_0))$ eine Tangente g anlegen, sodass f vollständig oberhalb von g verläuft.



Dass das geht sieht man folgendemaßen (illustriert im Bild): Man betrachtet unterhalb des Punktes $(x_0, f(x_0))$ (rot) auf dem Funktionsgraphen von f (schwarz) den Winkelbereich derjenigen Richtungen, die niemals über den den Funktionsgraphen hinausgehen (blau). Ist dieser Bereich kleiner als 180° (rechts im Bild), dann hat man einen Widerspruch zur Konvexität von f. Ist dieser Bereich größer 180° (Mitte) oder gleich 180° (links), dann gibt es mindestens eine Gerade durch $(x_0, f(x_0))$, die vermeidet über den Funktionsgraphen von f hinauszugehen.

(c) Sei $g(x)$ die Funktion aus (b) für $x_0 = \mathbb{E}[X]$. Dann gilt $f(x) \geq g(x)$ für alle $x \in D$ und $f(\mathbb{E}[X]) = g(\mathbb{E}[X])$. Weil g eine Gerade ist gibt es $a, b \in \mathbb{R}$ sodass $g(x) = ax + b$. Es folgt

$$\mathbb{E}[f(X)] \geq \mathbb{E}[g(X)] = \mathbb{E}[aX + b] = a\mathbb{E}[X] + b = g(\mathbb{E}[X]) = f(\mathbb{E}[X]).$$

(d) Weil $-f$ konvex ist folgt direkt aus (c):

$$\mathbb{E}[f(X)] = -\mathbb{E}[-f(X)] \overset{(c)}{\leq} -(-f(\mathbb{E}[X])) = f(\mathbb{E}[X]).$$

Aufgabe 2 – 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) $\Pr[|\text{result} - m| \leq \varepsilon m] \geq 1 - 2\exp(-\varepsilon^2 pm/3)$.

(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 1.

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)

Lösung 2

(a) Seien $X_1, \ldots, X_m \sim \text{Ber}(p)$ unabhängige Zufallsvariablen, wobei X_i angibt, ob das $i$te Element des Streams zu einer Erhöhung des Zählers Z führt. Dann ist $X := \sum_{i=1}^{m} X_i$ der Wert von Z nach dem letzten Update. Die Schätzung des Algorithmus für m ist somit result $= X/p$. Daher gilt:

$$\mathbb{E}[\text{result}] = \mathbb{E}[X/p] = \tfrac{1}{p}\mathbb{E}\Big[\sum_{i=1}^{m} X_i\Big] = \tfrac{1}{p}\sum_{i=1}^{m}\mathbb{E}[X_i] = \tfrac{1}{p}\sum_{i=1}^{m} p = m.$$

(b) Mit der genannten Chernoff Schranke ergibt sich

$$\begin{aligned}\Pr[|\text{result} - m| \geq \varepsilon m] &= \Pr[|X/p - m| \geq \varepsilon m] = \Pr[|X - mp| \geq \varepsilon mp] \\ &= \Pr[|X - \mathbb{E}[X]| \geq \varepsilon\mathbb{E}[X]] \leq 2\exp(-\varepsilon^2\mathbb{E}[X]/3) = 2\exp(-\varepsilon^2 mp/3).\end{aligned}$$

Die Behauptung ergibt sich durch Betrachten der Gegenwahrscheinlichkeit.

(c) Da Z monoton wächst ist der Speicherbedarf für Z ganz am Ende am größten, nämlich $\lceil \log_2(1 + X) \rceil$. Weil $f(x) = \log(1 + x)$ konkav auf $[0, \infty)$ ist, folgt mit der Jensenschen Ungleichung

$$\begin{aligned}\mathbb{E}[\text{space}] = \mathbb{E}[\lceil \log_2(1 + X) \rceil] &\leq \mathbb{E}[\log_2(1 + X)] + 1 \\ &\overset{\text{Jensen}}{\leq} \log_2(1 + \mathbb{E}[X]) + 1 = \log_2(1 + mp) + 1.\end{aligned}$$