Karlsruher Institut für Technologie

Stefan Walzer
ITI Prof. Sanders
Wintersemester 2023/2024

Übungsblatt 2 – The Power of Randomness

Randomisierte Algorithmik

Aufgabe 1 – Polynomgleichungen überprüfen

Sei $\mathbb{F}$ ein Körper, zum Beispiel $\mathbb{F} = \mathbb{Q}$. Gegeben sei eine Polynomgleichung, zum Beispiel:

$$(x+1)(x-2)(x+3)(x-4)(x+5)(x-6) \stackrel{?}{=} x^6 - 7x^3 + 25$$

(a) Argumentiere: Für den Fall, dass die Beispielgleichung *nicht* gilt, gibt es höchstens 6 Werte für x, sodass auf beiden Seiten das gleiche herauskommt.

(b) Beschreibe einen randomisierten Algorithmus der entscheidet ob eine Polynomgleichung gilt oder nicht. Dieser darf falsche Polynomgleichungen mit kleiner Wahrscheinlichkeit als korrekt akzeptieren. Was lässt sich über diese Wahrscheinlichkeit sagen?

Lösung 1

(a) Die Gleichung hat die Form $f(x) = g(x)$ und lässt sich umstellen zu $f(x) - g(x) = 0$. Falls die Gleichung nicht gilt ist $f(x) - g(x)$ ein Polynom von Grad $0 \leq d \leq 6$. Ein solches Polynom hat höchstens 6 Nullstellen. Also gibt es höchstens 6 Werte für x für die $f(x) - g(x) = 0$ und damit $f(x) = g(x)$ gilt.

(b) Betrachten wir zunächst den Fall $|\mathbb{F}| < \infty$. Wir wählen dann $X \sim \mathcal{U}(\mathbb{F})$ zufällig und setzen X in die Gleichung ein. Falls die Polynomgleichung allgemein stimmt, dann auch für dieses X. Falls sie nicht stimmt, dann können wir wie in (a) den maximalen Grad d betrachten, der auf beiden Seiten vorkommt. Es gibt höchstens d Elemente von $\mathbb{F}$, für die auf beiden Seiten das gleiche herauskommt, und die Wahrscheinlichkeit, dass wir eines davon gewählt haben ist höchstens $d/|\mathbb{F}|$.

Falls $|\mathbb{F}| = \infty$ gibt es keine Gleichverteilung auf $\mathbb{F}$. Das ist einerseits etwas lästig, andererseits können wir X gleichverteilt aus einer großen Teilmenge $S \subseteq \mathbb{F}$ wählen. Dann wird die obere Schranke für die Fehlerwahrscheinlichkeit mit $d/|S|$ sogar beliebig klein.

Bemerkung: Es handelt sich um einen false-biased Monte-Carlo-Algorithmus (siehe Vorlesung zu Probability Amplification).

Aufgabe 2 – Matrixprodukte überprüfen[1]

Seien $\mathbb{F}$ ein Körper, $n \in \mathbb{N}$.

(a) Zeige: Falls $C, C' \in \mathbb{F}^{n \times n}$ zwei verschiedene Matrizen sind und $v \in \{0,1\}^n$ uniform zufällig gewählt wird, dann gilt: $\Pr[C \cdot v \neq C' \cdot v] \geq \frac{1}{2}$.

(b) Beschreibe einen Algorithmus der bei Eingabe $A, B, C \in \mathbb{F}^{n \times n}$ ein Bit X ausgibt mit $X = 1$ falls $A \cdot B = C$ und $\Pr[X = 1] \leq 1/2$ falls $A \cdot B \neq C$. Der Algorithmus soll nur $O(n^2)$ Körperoperationen durchführen.

Lösung 2

(a) Seien $C_1, \ldots, C_n \in \{0,1\}^n$ und $C'_1, \ldots, C'_n \in \{0,1\}^n$ die Spalten von C und C' sowie $v_1, \ldots, v_n \in \{0,1\}$ die Einträge von v. Sei ferner $i \in [n]$ ein Index mit $C_i \neq C'_i$, der nach Vorraussetzung existieren muss. Wir können nun schreiben:

$$D := C \cdot v - C' \cdot v = \underbrace{\sum_{\substack{j=1 \\ j \neq i}}^{n} (C_j - C'_j) \cdot v_j}_{w} + (C_i - C'_i) \cdot v_i.$$

Wir stellen uns nun vor, dass zunächst alle Einträge von v bis auf den $i$ten Eintrag gewählt werden und nur v_i noch zufällig ist. Nun gibt es zwei Fälle.

Fall 1. $w = 0$. In dem Fall führt $v_i = 1$ zu $D = C_i - C'_i \neq 0^n$.

Fall 2. $w \neq 0$. In dem Fall führt $v_i = 0$ zu $D = w \neq 0^n$.

In beiden Fällen führt also mindestens eine der beiden Möglichkeiten für v_i zu $D \neq 0^n$. Also gilt auch insgesamt $\Pr[C \cdot v \neq C' \cdot v] = \Pr[D \neq 0^n] \geq \frac{1}{2}$.

(b) Wir verwenden (a) mit $C' = A \cdot B$.

sample $v \leftarrow \mathcal{U}(\{0,1\}^n)$
$w_1 \leftarrow C \cdot v$
$w_2 \leftarrow A \cdot B \cdot v$ // Berechnungsreihenfolge: $A \cdot (B \cdot v)$
return $X = \mathbb{1}_{w_1 = w_2}$

Offenbar ist $X = 1$ garantiert falls $A \cdot B = C$. Falls $A \cdot B \neq C$ gilt nach (a) $\Pr[X = 1] = \Pr[C \cdot v = C' \cdot v] \leq \frac{1}{2}$.

Die drei Matrix-Vektor Produkte lassen sich jeweils in $O(n^2)$ Körperoperationen ausführen. Insbesondere ist das schneller als ein Matrix-Matrix Produkt das (mit einem naiven Algorithmus) $\Omega(n^3)$ Körperoperationen benötigt.

Bemerkung: Es handelt sich um einen false-biased Monte-Carlo-Algorithmus (siehe Vorlesung zu Probability Amplification).

[1] Bekannt als Algorithmus von Freivald.

Aufgabe 3 – Deterministische Auswertung von $\bar{\wedge}$-Bäumen

Sei A ein deterministischer Algorithmus der einen Bitvektor $I \in \{0,1\}^n$ als Eingabe erhält (mit $n = 2^d$) und den Wert des vollständigen balancierten $\bar{\wedge}$-Baums berechnet, dessen Blätter gemäß I beschriftet sind. Zeige: Es gibt eine Eingabe $I_A \in \{0,1\}^n$, sodass A jede Blattbeschriftung inspizieren muss.

Lösung 3

Wir sind ein Widersacher des Algorithmus und legen den Wert eines Blattes erst dann fest, wenn es angefragt wird. Wir stellen sicher, dass er alle Blätter anfragen muss. Die Belegung der Blätter die sich ergibt ist dann die Worst-Case Anfrage I_A für A.

Wir zeigen die Aussage per Induktion. Für $d = 1$ (Blatt = Wurzel) ist nichts zu zeigen. Der Algorithmus muss das einzige Blatt natürlich anfragen.

Für $d > 1$ liegen zwei Unterbäume der Tiefe $d-1$ vor. In beiden benutzen wir die Strategie, die sich aus der Induktion ergibt. Damit kann sich das Ergebnis eines Unterbaumes erst dann herausstellen, wenn alle 2^{d-1} Blätter angefragt wurden. Weil das letzte Blatt eine Rolle für das Ergebnis des Unterbaums gespielt hat, können wir sogar im letzten Schritt noch aussuchen, dass der Unterbaum insgesamt zu 1 auswertet. Damit ist das Gesamtergebnis an der Wurzel $1 \bar{\wedge} b$ wobei b das Ergebnis des anderen Unterbaumes ist. Das muss der Algorithmus also auch noch bestimmen, und damit auch im zweiten Unterbaum alle Blätter anfragen.