Karlsruher Institut für Technologie

Stefan Walzer
ITI Prof. Sanders
Wintersemester 2023/2024

Übungsblatt 4 – Probability Amplification

Randomisierte Algorithmik

Aufgabe 1 – Probability Amplification mit zweiseitigem Fehler

Angenommen ein Monte-Carlo Algorithmus A beantwortet ein Entscheidungsproblem mit Wahrscheinlichkeit $1/2 + \varepsilon$ korrekt und sonst falsch (für ein $\varepsilon > 0$). Sei A' der Algorithmus der t mal unabhängig A ausführt und sich für die häufigere Antwort entscheidet. Zeige, dass die Fehlerwahrscheinlichkeit von A' höchstens $e^{-2t\varepsilon^2}$ ist.

Hinweise: Für die Rechnung könnte nützlich sein: $\sum_{i=0}^{t} \binom{t}{i} = 2^t$.

Lösung 1

Sei $I \in \{0, \dots, t\}$ die Anzahl der Ausführungen, die das richtige Ergebnis liefern. Damit A' falsch antwortet muss $I \leq t/2$ gelten. Die Lösung ergibt sich durch geschicktes Abschätzen:

$$\begin{aligned}
\Pr[I \leq t/2] &= \sum_{i=0}^{\lfloor t/2 \rfloor} \Pr[I \leq i] = \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} (\tfrac{1}{2} + \varepsilon)^i (\tfrac{1}{2} - \varepsilon)^{t-i} \leq \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} (\tfrac{1}{2} + \varepsilon)^{t/2} (\tfrac{1}{2} - \varepsilon)^{t/2} \\
&\leq (\tfrac{1}{4} - \varepsilon^2)^{t/2} \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{t}{i} \leq (\tfrac{1}{4} - \varepsilon^2)^{t/2} \sum_{i=0}^{t} \binom{t}{i} = (\tfrac{1}{4} - \varepsilon^2)^{t/2} \cdot 2^t \\
&= (1 - 4\varepsilon^2)^{t/2} \leq (e^{-4\varepsilon^2})^{t/2} = e^{-2t\varepsilon^2}.
\end{aligned}$$

Aufgabe 2 – Pfadfinder

Gegeben sei ein ungerichteter Graph G mit n Knoten und m Kanten. Gesucht ist ein Pfad der Länge k, der keinen Knoten mehrfach besucht. Der naive Brute-Force Ansatz hat eine Laufzeit von $O(n^k)$. Wir betrachten einen einfachen, randomisierten Ansatz, der wie folgt funktioniert. Im ersten Schritt wird jedem Knoten v ein Label $L(v)$ zufällig gleichverteilt aus $\{1, \dots, k\}$ zugewiesen. Im zweiten Schritt wird von jedem Knoten v mit $L(v) = 1$ eine modifizierte Breitensuche gestartet, bei der ein Knoten w nur dann von einem Knoten u entdeckt werden kann, falls $L(u) = L(w) + 1$ gilt. Wird ein Knoten mit Label k entdeckt wird ein Pfad der Länge k konstruiert und ausgegeben.

(a) Zeige, dass der Algorithmus mit Wahrscheinlichkeit $1/k^k$ einen Pfad der Länge k findet, falls einer existiert.

(b) Verbessere die Erfolgswahrscheinlichkeit auf $1 - 1/n$ mit Probability Amplification. Welche Gesamtlaufzeit ergibt sich?

Bemerkung: Die Idee heißt "Color Coding". Es gibt noch raffiniertere Varianten.

Lösung 2

(a) Sei $P = (v_1, \ldots, v_k)$ ein Pfad der Länge k in G. Mit Wahrscheinlichkeit $1/k^k$ wird für jedes $i \in [k]$ der Knoten v_i mit Farbe i gefärbt. In dem Fall wird die Breitensuche die Knoten $v_1, \ldots, v_k$ alle erreichen (entlang von P oder entlang eines andere Pfades) und somit einen Pfad der Länge k finden.

(b) Wir führen den Algorithmus $t = k^k \cdot \ln n$ mal aus. Die Erfolgswahrscheinlichkeit beträgt dann wie gewünscht:

$$1 - (1 - k^{-k})^t \geq 1 - (e^{-k^{-k}})^t = 1 - e^{-\ln n} = 1 - \tfrac{1}{n}.$$

Da n Breitensuchen in Zeit $O(nm)$ durchgeführt werden können, ergibt sich insgesamt eine Laufzeit von $O(nm \cdot k^k \cdot \ln n)$.

Aufgabe 3 – Bonus: Random-Walk Löser für 3-SAT

Sei $\varphi(x_1, \ldots, x_n)$ eine 3-SAT Formel mit n Variablen und m Klauseln. Wir suchen eine Lösung mit folgendem Algorithmus:

Algorithm randomWalkSolver(φ)**:**
- sample $x_1, \ldots, x_n \sim \mathrm{Ber}(1/2)$
- **for** $k = 1$ **to** $n/2$ **do**
  - **if** $\varphi(x_1, \ldots, x_n) = 1$*)* **then**
    - **break**
  - sei C eine beliebige unerfüllte Klausel von φ
  - sample $j \sim \mathcal{U}(\{1, 2, 3\})$
  - sei x_i die $j$te Variable in C
  - $x_i \leftarrow 1 - x_i$
- **if** $\varphi(x_1, \ldots, x_n) = 1$*)* **then**
  - **return** $(x_1, \ldots, x_n)$
- **return** $\perp$

Falls φ unerfüllbar ist, gilt offenbar randomWalkSolver(φ) $= \perp$. Andernfalls sei x^* eine Lösung. Zeige:

(a) Mit Wahrscheinlichkeit $\geq 1/2$ stimmt die initial gesampelte Belegung $(x_1, \ldots, x_n)$ an mindestens $n/2$ Stellen mit x^* überein.

(b) Falls $\varphi(x_1, \ldots, x_n) = 0$ so führt eine Iteration der Schleife mit Wahrscheinlichkeit $1/3$ dazu, dass eine weitere Variable von $(x_1, \ldots, x_n)$ so wie in x^* gesetzt wird.

(c) Verwende Probability Amplification um einen Algorithmus mit Erfolgswahrscheinlichkeit $1 - 1/n$ zu erhalten. Was ist die Gesamtlaufzeit?

Lösung 3

(a) Für jede "schlechte" Belegung mit weniger als $n/2$ Übereinstimmungen hat die invertierte Belegung mehr als $n/2$ Übereinstimmung. Damit machen die "schlechten" Belegungen höchstens einen Anteil von $1/2$ aus. (Die Belegungen mit exakt $n/2$ Übereinstimmungen führen bei geradem n zu einem Bias zu unseren Gunsten).

(b) In der unerfüllten Klausel C, die im Schleifendurchlauf betrachtet wird, kommt mindestens eine Variable vor, die in x^* anders belegt ist als in $(x_1, \ldots, x_n)$, schließlich erfüllt x^* ja C. Mit Wahrscheinlichkeit $1/3$ wählen wir diese Variable aus.

(c) Zunächst ergibt sich aus (a) und (b), dass die Erfolgswahrscheinlichkeit bei einmaliger Ausführung mindestens $\frac{1}{2} \cdot 3^{-n/2}$ beträgt (intuitiv: wir starten in der Nähe von x^* und bewegen uns nur darauf zu). Wiederholen wir den Algorithmus $t = 2 \cdot 3^{n/2} \cdot \ln n$ mal, so ergibt sich eine Erfolgswahrscheinlichkeit von

$$1 - (1 - \tfrac{1}{2} \cdot 3^{-n/2})^t \geq 1 - (e^{-\frac{1}{2} \cdot 3^{-n/2}})^t = 1 - e^{-\ln n} = 1 - \tfrac{1}{n}.$$

Wenn m die Anzahl von Klauseln in φ ist, dann hat randomWalkSolver eine Laufzeit von $O(nm)$. Insgesamt ergibt sich $O(3^{n/2} \cdot nm)$. Das ist potentiell besser als ein naiver Algorithmus mit Laufzeit $O(2^n)$.