Karlsruher Institut für Technologie

Maximilian Katzmann
Hans-Peter Lehmann
Stefan Walzer

Übungsblatt 13 – Aktivsession

Randomisierte Algorithmik – Wintersemester 2023/2024

Folgende Aufgaben werden in der Aktiv-Session am 1.2.2024 gemeinsam bearbeitet. Gib eine Lösung zu einer *der beiden Aufgaben (deine Wahl) wie gewohnt bis zum 8.2.2024 über Ilias ab.*

Aufgabe 1 – Schnitte Schätzen mit Bloomfiltern

Seien $n, m, k \in \mathbb{N}$. Alice und Bob wollen abschätzen, wie ähnlich ihr Musikgeschmack ist. Seien X die n Lieblingslieder von Alice und Y die n Lieblingslieder von Bob. Zu schätzen ist $\gamma := \frac{|X \cap Y|}{n} \in [0, 1]$. Beide gehen folgendermaßen vor.

- Alice konstruiert einen Bloomfilter $A[1..m] \in \{0,1\}^m$ für X unter Verwendung von k Hashfunktionen $h_1, \ldots, h_k$.
- Bob konstruiert einen Bloomfilter $B[1..m] \in \{0,1\}^m$ für Y unter Verwendung *derselben* k Hashfunktionen.
- Alice und Bob tauschen ihre Filter aus und berechnen $\delta := \frac{|\{i \in [m] | A[i] \neq B[i]\}|}{m}$.
- Alice und Bob berechnen basierend auf δ eine Schätzung $\bar{\gamma}$ für γ.

Löse folgende Teilaufgaben:

(a) Diskutiere: Welche Vor- und Nachteile könnte das Verfahren im Vergleich zum direkten Austausch von X und Y haben?

(b) Gewinne Intuition: Welche Werte von δ erwartest du (in etwa) für die Extremfälle, in denen $\gamma = 1$ bzw. $\gamma = 0$ gilt?
Hinweis: Du darfst hier und im Folgenden davon ausgehen, dass den Bloomfiltern eine „optimale“ Konfiguration mit $\alpha k = \ln(2)$ zugrundegelegt wurde.

(c) Berechne $\mathbb{E}[\delta]$ als Funktion von γ. Du darfst hierbei Terme niederer Ordnung unter den Tisch fallen lassen, also z.B. $(1 - \frac{1}{m})^m \approx e^{-1}$ schreiben, ohne ein $o(1)$ mitzuführen.
Hinweis: Zunächst scheint es, als könnten andere Parameter (z.B. $n, m, k, \alpha, \varepsilon$) auch eine Rolle spielen. Deren Einfluss verschwindet aber in Termen niederer Ordnung.

(d) Diskutiere: Welche Konzentrationsschranke eignet sich, um zu beweisen, dass δ mit hoher Wahrscheinlichkeit nahe an $\mathbb{E}[\delta]$ liegt?

(e) Stelle die Gleichung aus (c) um, sodass ersichtlich wird, wie eine Schätzung $\bar{\gamma}$ für γ aus δ berechnet werden kann.

(f) Spekuliere: Welche Rolle spielt die Wahl von k (bzw. von ε) im vorliegenden Kontext?

Lösung 1

(a)
- Vorteil: Der Platzverbrauch ist unter Umständen geringer.
- Nachteil: Wir können zwar erreichen, dass $\bar{\gamma}$ mit hoher Wahrscheinlichkeit in der Nähe von γ liegt, aber wir können γ nicht fehlerfrei berechnen.
- Vorteil: Die Elemente der Mengen X und Y werden nicht bekanntgegeben, d.h. Alice kann für jedes $x \in X$ abstreiten dass $x \in X$ gilt, ohne dass jemand sie der Lüge überführen kann.

(b) Für $\gamma = 1$ gilt offensichtlich $A[i] = B[i]$ für alle i und damit $\delta = 0$. Für $\gamma = 0$ haben die beiden Bloomfilter nichts miteinander zu tun und sind unabhängig. Laut Vorlesung sind in einem Bloomfilter mit $\alpha k = \ln(2)$ etwa die Hälfte der Einträge 1 und die andere Hälfte ist 0. Plausibel ist also, dass für alle $i \in [m]$ gilt: $\Pr[A[i] \neq B[i]] \approx \Pr_{C,D\sim\mathrm{Ber}(1/2)}[C \neq D] = 1/2$. Wir erwarten also $\delta \approx 1/2$.

(c) Wir schreiben kurz $h(z) := \{h_1(z), \ldots, h_k(z)\}$. Beachte: Ereignisse, die sich auf verschiedene Schlüssel oder verschiedene Hashfunktionen beziehen können wir wegen Unabhängigkeit „auseinanderziehen“. Im Folgenden nutzen wir x_0 für ein beliebiges Element in X und y_0 für ein beliebiges Element in $Y \setminus X$.

$$
\begin{aligned}
\mathbb{E}[\delta] &= \frac{\mathbb{E}[|\{i \in [m] \mid A[i] \neq B[i]\}|]}{m} = \frac{1}{m}\sum_{i=1}^{m} \Pr[A[i] \neq B[i]] = \Pr[A[i_0] \neq B[i_0]] \\
&= \Pr[(A[i_0] = 0 \wedge B[i_0] = 1) \vee (A[i_0] = 1 \wedge B[i_0] = 0)] = 2\Pr[A[i_0] = 0 \wedge B[i_0] = 1] \\
&= 2\Pr[\forall x \in X : i_0 \notin h(x) \wedge \exists y \in Y \setminus X : i_0 \in h(y)] \\
&= 2\Pr[\forall x \in X : i_0 \notin h(x)] \cdot \Pr[\exists y \in Y \setminus X : i_0 \in h(y)] \\
&= 2\Pr[\forall x \in X : i_0 \notin h(x)] \cdot (1 - \Pr[\forall \in Y \setminus X : i_0 \notin h(y)]) \\
&= 2\Pr[i_0 \notin h(x_0)]^{|X|} \cdot (1 - \Pr[i_0 \notin h(y_0)]^{|Y\setminus X|}) \\
&= 2\Pr[i_0 \neq h_1(x_0)]^{k|X|} \cdot (1 - \Pr[i_0 \neq h_1(y_0)]^{k|Y\setminus X|}) \\
&= 2(1 - \tfrac{1}{m})^{k|X|} \cdot (1 - (1 - \tfrac{1}{m})^{k|Y\setminus X|}) = 2(1 - \tfrac{1}{m})^{kn} \cdot (1 - (1 - \tfrac{1}{m})^{k(1-\gamma)n}) \\
&= 2(1 - \tfrac{1}{m})^{k\alpha m} \cdot (1 - (1 - \tfrac{1}{m})^{k(1-\gamma)\alpha m}) \\
&\approx 2e^{-k\alpha} \cdot (1 - e^{-k(1-\gamma)\alpha}) = 2e^{-\ln(2)} \cdot (1 - e^{-(1-\gamma)\ln(2)}) = 1 - \left(\tfrac{1}{2}\right)^{(1-\gamma)}.
\end{aligned}
$$

(d) Die Methode der beschränkten Differenzen bzw. McDiarmids Ungleichung ist hier geeignet. Die $k \cdot |X \cup Y|$ relevanten Hashwerte sind alle unabhängig. Verändert man einen davon dann ändert sich der Wert von δ um höchstens $\pm\frac{1}{m}$. Die Analyse der Vorlesung bzgl. der Konzentration von Z (Anzahl Nuller) überträgt sich problemlos.

(e) Aus (c) haben wir $\mathbb{E}[\delta] = 1 - (\frac{1}{2})^{(1-\gamma)}$. Wir entfernen das „$\mathbb{E}$“ (weil wir nur δ zur Verfügung haben, nicht $\mathbb{E}[\delta]$) und ersetzen γ durch $\bar{\gamma}$ (weil wir also nicht γ ausrechnen können sondern nur eine Schätzung dafür). Umstellen von $\delta = 1 - (\frac{1}{2})^{(1-\bar{\gamma})}$ ergibt: $\bar{\gamma} = 1 - \log_2(1/(1-\delta))$.

(f) Je größer k ist, desto stärker wird die Konzentrationsschranke und desto besser wird entsprechend die Schätzung $\bar{\gamma}$. Es spricht aber wenig dagegen einfach $k = 1$ zu verwenden. Es ist sogar denkbar $k \in (0, 1)$ zu wählen mit der Bedeutung, dass ein Schlüssel nur mit Wahrscheinlichkeit k eine Position zugeteilt bekommt und mit Wahrscheinlichkeit $1 - k$ einfach verworfen wird. Diese Zufallsentscheidungen müssten widerum durch eine Hashfunktion getroffen werden, die Alice und Bob beide kennen. Mit $k = \Theta(\frac{1}{n})$ könnte man einen Speicherbedarf erreichen der nicht mehr von n abhängt. Ähnlich wie bei Approximationsalgorithmen könnte man einen relativen Fehler und eine Fehlerwahrscheinlichkeit einführen und ausrechen wie groß k in Abhängigkeit dieser beiden Parameter gewählt werden müsste.

Aufgabe 2 – Cuckoo Hashing: Verwandte Fragen und Modelle

Aus mathematischer Sicht ist es oft nebensächlich und ablenkend Cuckoo Hashing mit zwei getrennten Tabellen aufzuziehen. Betrachten wir daher stattdessen folgende Variante:

Es gibt *eine* Tabelle der Größe m, es gibt *zwei* Hashfunktionen $h_0, h_1 : D \to [m]$ und ein Schlüssel $x \in D$ darf entweder in $h_0(x)$ oder in $h_1(x)$ platziert werden.

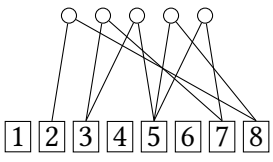
(a) Passe die insert-Methode aus der Vorlesung auf diese Variante an. Welcher Nachteil ergibt sich für die Anzahl der nötigen Hashfunktionsevaluationen?

Betrachte den bipartiten Graphen $B = (V_0, V_1, E_B)$ und den gewöhnlichen Graphen $C = (V, E)$, die für eine gegebene Schlüsselmenge $S \subseteq D$ folgendermaßen definiert sind.

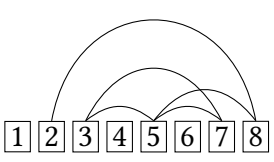
$$B = (S, [m], \{(x, h_0(x)) \mid x \in S\} \cup \{(x, h_1(x)) \mid x \in S\})$$
$$C = ([m], \{\{h_0(x), h_1(x)\} \mid x \in S\})$$

Beachte: In C sind sowohl Multikanten als auch Schleifen möglich. Beispiel: Für eine Schlüsselmenge $S = \{x_1, \ldots, x_5\}$ und Hashwerte wie folgt ergeben sich für B und C die Graphen wie illustriert.

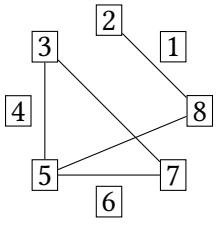
	$h_0(x)$	$h_1(x)$
x_1	2	8
x_2	7	3
x_3	3	5
x_4	5	8
x_5	7	5



B



C



auch C

(b) Zeige dass folgende Aussagen äquivalent sind.

(i) Alle Schlüssel aus S können in der Hashtabelle platziert werden.

(ii) Es gibt ein Matching der Größe $n = |S|$ in B.

(iii) C ist ein Pseudowald.

(iv) Man kann jede Kante aus C mit einer Richtung versehen, sodass jeder Knoten im entstandenen gerichteten Graphen $\vec{C}$ Eingangsgrad höchstens 1 hat.

Hinweis: Ein Pseudowald ist ein Graph in der jede Komponente ein Pseudobaum ist. Für einen Pseudobaum gibt es mehrere äquivalente Definitionen:

- Zusammenhängend und #Kanten ≤ #Knoten.
- Zusammenhängend und höchstens ein Kreis.
- Ein Baum plus bis zu eine zusätzliche Kante.

Siehe auch: `https://en.wikipedia.org/wiki/Pseudoforest`

Sei $G_{m,n}$ ein Zufallsgraph im Erdős-Renyi Modell mit m Knoten und n Kanten. Das heißt unter allen $\binom{\binom{m}{2}}{n}$ Graphen mit Knotenmenge $[m]$ und n Kanten ist $G_{m,n}$ uniform zufällig gewählt. Der Graph C hat große Ähnlichkeiten mit $G_{m,n}$.

(c) Mache dir die feinen Unterschiede zwischen der Verteilung von $G_{m,n}$ und C klar. Überlege dir zum Beispiel Urnenmodelle für die beiden Verteilungen. Wieviele Bälle sind in der Urne? Was bedeuten sie? Sampeln wir mit oder ohne Zurücklegen?

Lösung 2

(a) Weil es nur noch eine Tabelle gibt, ist nicht unmittelbar ersichtlich, ob ein Schlüssel, den man an einer Position i vorfindet, gemäß Funktion h_0 oder h_1 platziert ist. Um seine Alternativposition herauszubekommen, muss man also im Zweifelsfall beide Hashfunktionen ausprobieren und schauen welche einen Hashwert $\neq i$ liefert. Daher ist die erwartete Anzahl von Hashfunktionsauswertungen größer.

```
Algorithm insert(x):
    j ← h_0(x) // wo x gleich hin soll
    for i = 0 to limit do
        swap(x, T[j])
        if x = ⊥ then
            return SUCCESS
        // finde Alternativposition für x:
        j' ← h_0(x)
        if j = j' then
            j ← h_1(x)
        else
            j ← j'
    return FAILURE
```

(b) Wir machen einen Ringschluss:

(i) ⇒ (ii). Wenn $x_1, \ldots, x_n$ an Positionen $i_1, \ldots, i_n$ unterkommen, dann betrachten wir die Kantenmenge $M = \{(x_j, i_j) \mid j \in [n]\}$ in B. Diese bildet ein Matching: Dass

Schüsselknoten nicht doppelt verwendet werden ist offensichtlich, für die Tabellenknoten folgt dies aus der Kollisionsfreiheit der Platzierung. Also ist M ein Matching der Größe n.

(ii) ⇒ (iii). Sei K eine Zusammenhangskomponente von C und K_B die zugehörige Zusammenhangskomponente von B (es dürfte klar sein, was „zugehörig“ hier heißt, oder?). Sei M ein Matching der Größe n in B. Beachte: Jeder Schlüssel-Knoten in B hat durch M einen Tabellen-Knoten als Matchingpartner. Daher gilt für die Anzahl n' der Schlüssel-Knoten in K_B und die Anzahl m' der Tabellen-Knoten in K_B dass $n' \leq m'$. Beachte: n' ist die Anzahl von Kanten in K und m' die Anzahl von Knoten in K. Weil K zusammenhängend ist, gilt ferner $n' \geq m' - 1$. Es gibt also nur zwei Möglichkeiten: Für $n' = m'$ ist K ein Pseudobaum und für $n' = m' - 1$ ist K sogar ein Baum.

(iii) ⇒ (iv). Wir können die Zusammenhangskomponenten von C getrennt betrachten. Jede ist ein Baum oder ein Pseudobaum mit einem Kreis. Für Bäume können wir eine beliebige Wurzel auszeichnen und dann alle Kanten im Baum „von der Wurzel weg“ richten. In diesem gerichteten Baum hat dann jeder Knoten Eingangsgrad 1, außer die Wurzel, die Eingangsgrad 0 hat. Für Pseudobäume mit einem Kreis können wir den Kreis in einer beliebigen Richtung durchlaufen und die Kanten darauf entsprechend richten. Alle weiteren Kanten liegen auf „Abzweigungen“ vom Kreis und können „davon weg“ gerichtet werden. In diesem gerichteten Pseudobaum hat jeder Knoten exakt Eingangsgrad 1.

(iv) ⇒ (i). Wenn eine Kanten e, die von einem Schlüssel x herkommt, zu einem Knoten $i \in \{h_0(x), h_1(x)\}$ gerichtet wird, dann platzieren wir x in Tabellenposition i. So wird jeder Knoten an einer für ihn erlaubten Position platziert. Kollisionen gibt es nicht, nach der „Eingangsgrad ≤ 1“ Bedingung.

(c) Zwei auffällige Unterschiede sind, dass C Doppelkanten haben kann (wenn für $x \neq y \in S$ gilt dass $\{h_0(x), h_1(x)\} = \{h_0(y), h_1(y)\}$) sowie Schleifen haben kann (wenn für $x \in S$ gilt $h_0(x) = h_1(x)$). Für $G_{m,n}$ ist das nicht möglich. Urnenmodelle für die beiden Situationen wäre:

Für C: Präpariere eine Urne mit m Bällen. Sample $2n$ mal aus der Urne mit zurücklegen, das ergibt $i_1, \ldots, i_{2n} \in [m]$. Form dann die n Kanten $\{i_1, i_2\}, \{i_3, i_4\}, \ldots, \{i_{2n-1}, i_{2n}\}$.

Für $G_{m,n}$: Präpariere eine Urne mit $\binom{m}{2}$ Bällen – einem Ball für jede zweielementige Teilmenge von $[m]$. Ziehe dann n mal *ohne Zurücklegen* aus der Urne. Nehme die entsprechenden Kanten in $G_{m,n}$ auf.