Internet-Algorithmen

Sommersemester 2014

Prof. Dr. Georg Schnitger
Hannes Seiwert

Arbeitsgruppe Theoretische Informatik, Institut für Informatik

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Übungsblatt 7

Ausgabe: 23.06.2014
Abgabe: 30.06.2014 vor Vorlesungsbeginn

Hinweis: Alle Lösungen sind ordentlich und in lesbarer Schrift zu verfassen. Fasse dich kurz, beschränke deine Erläuterungen und Rechnungen auf die wesentlichen Punkte.

Hinweis: Die Gesamtpunktzahl beträgt 24 Punkte. Darüber hinaus erworbene Punkte werden als Bonuspunkte angerechnet.

Aufgabe 7.1. (3+4+3) *Sample and Count*

Gegeben sei ein Datenstrom $x_1, \ldots, x_n$. Wir beschreiben ein probabilistisches Verfahren zur Bestimmung von Heavy Hitters, den Algorithmus Sample-and-Count. Die Länge n des Datenstroms sei vorher bekannt.

(1) Initialisiere eine Stichprobe $S = \emptyset$. Für jedes Element x des Datenstroms führe die folgende Fallunterscheidung durch:

(i) Falls $x \notin S$, dann füge x zu S mit Wahrscheinlichkeit $\frac{r}{n}$ zu S hinzu. Initialisiere einen Zähler C_x für x mit dem Wert 1.
// r wird später bestimmt.

(ii) Falls $x \in S$, dann erhöhe den Zähler C_x von x um eins.

(2) Gib alle Schlüssel in S aus, deren Zählerwert mindestens $(\theta - \varepsilon) \cdot n$ beträgt.
// θ und ε werden vom Anwender vorgegeben. Es muss $\varepsilon \leq \theta$ gelten.

a) Als Heavy Hitter definieren wir alle Elemente u, die mit Häufigkeit $a_u \geq \theta\, n$ auftreten.

Sei y ein Heavy Hitter. **Berechne** eine obere Schranke für die Wahrscheinlichkeit p_y, dass y vom Algorithmus in Schritt (2) *nicht* ausgegeben wird, in Abhängigkeit von r und ε.

b) Offensichtlich gibt es höchstens $1/\theta$ viele Heavy Hitter. **Berechne** eine obere Schranke für die Wahrscheinlichkeit p, dass *irgendein* Heavy Hitter *nicht* ausgegeben wird, in Abhängigkeit von r, ε und θ.
Bestimme anschließend r, so dass $p \leq \delta$ für ein gegebenes $\delta \in (0,1)$ gilt und **zeige**, dass die erwartete Stichprobengröße $|S|$ durch $\mathcal{O}\left(\frac{1}{\varepsilon} \ln\left(\frac{1}{\theta\,\delta}\right)\right)$ beschränkt ist.

c) **Beschreibe** die Vor- und Nachteile dieses Verfahrens gegenüber dem Zähleralgorithmus.

Aufgabe 7.2. (4+4+4)

Häufigkeiten mit Hashing

Der Datenstrom $D = x_1, \ldots, x_n$ mit Elementen $x_i \in U$ sei gegeben. Wir wählen zufällig k unabhängige Hashfunktionen $h_1, \ldots, h_k : U \to \{-1, +1\}$ aus und richten k anfänglich auf Null gesetzte Zähler $Z_1, \ldots, Z_k$ ein.

> Für jedes Element x_i des Datenstroms und für jedes $j \in [k]$ setze $Z_j := Z_j + h_j(x_i)$. Anschließend geben wir für jedes Element $u \in U$ den Wert $H'(u) = \frac{1}{k} \cdot \sum_{i=1}^{k} Z_i \cdot h_i(u)$ als Schätzung der Häufigkeit aus.

Sei $H(u)$ die tatsächliche Häufigkeit des Schlüssels u.

a) **Zeige**, dass $E[H'(u)] = H(u)$ gilt.

b) Sei $||H|| := \sqrt{\sum_{y \in U} H(y)^2}$. **Zeige**, dass $\mathrm{Var}(H'(u)) = \frac{1}{k} \cdot \sum_{y \in U \setminus \{u\}} H(y)^2 \leq \frac{1}{k} ||H||^2$ gilt.

c) **Zeige**, dass eine Schätzung $H''(u)$ mit Speicherplatzkomplexität $\mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \ln\left(\frac{1}{\delta}\right)\right)$ berechnet werden kann, so dass $\mathrm{prob}\left[\, |H''(u) - H(u)| \geq \varepsilon \cdot ||H|| \,\right] \leq \delta$ gilt.

Hinweis: Wähle dazu ein geeignetes k und wende das Boosting-Verfahren an.

Aufgabe 7.3. (6)

Einsen zählen im gewichteten Zeitfenster

Auf einem Datenstrom $D = x_1, \ldots, x_n$ mit Elementen $x_i \in \{0, 1\}$ wollen wir die Einsen in einem gleitenden Zeitfenster zählen. Anders als in der Vorlesung habe dieses Zeitfenster allerdings kein festes Ende, sondern stattdessen eine exponentiell abfallende Gewichtung: Je älter eine betrachtete 1 ist, desto weniger soll sie zu unserer Zählung beitragen.

Konkret: Es sei $q \in (0, 1)$ fest. Im Zeitfenster Z_t soll eine 1, die zum Zeitpunkt $k \leq t$ auftrat, mit Gewicht q^{t-k} gezählt werden.

Beispiel: Wenn $q = 1/2$ gilt und bisher der Datenstrom $0, 1, 1, 0, 1, 0, 1$ betrachtet wurde, soll der Zählerstand $0 + (1/2)^5 + (1/2)^4 + 0 + (1/2)^2 + 0 + (1/2)^0 = 43/32$ betragen.

Entwirf einen deterministischen Algorithmus, der mit möglichst geringem Speicherbedarf die Einsen im Zeitfenster Z_t für alle $t \in [n]$ zählt.