

Approximate Frequency Counts over Data Streams

Gurmeet Singh Manku Rajeev Motwani

Näherungsweise
Häufigkeitszählung in Datenströmen

Seminarvortrag von Marco Möller

Wofür ist das gut?

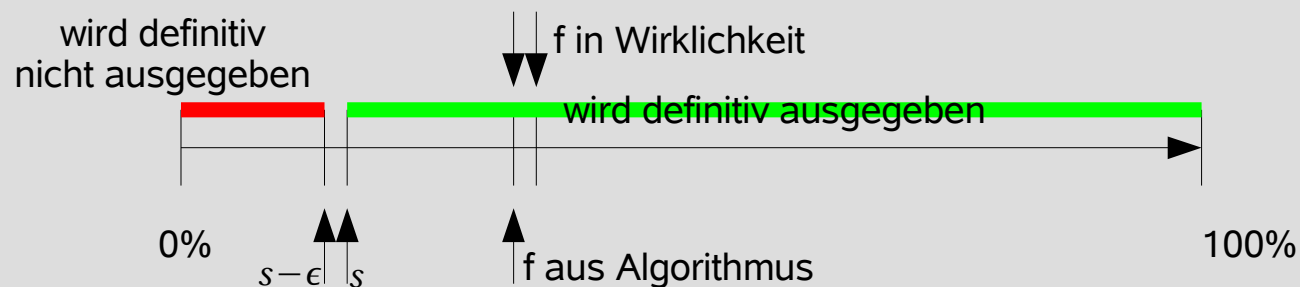
- Was sind die Top Ten der Suchanfragen bei g**gle.de ?
- Lernen von Assoziationsregeln:
 - Kunden, die das neue MS Office bestellen, bestellen auch häufig zusätzlichen RAM
- IP Traffic Mangagement:
 - finden von viel genutzten Links in Netzwerken um Routen danach zu optimieren
 - „Denial of Service“ Attacken aufdecken

Worum geht's?

- In einem Datenstrom der Länge N sollen alle Elemente die häufiger als $s \cdot N$ mit $s \in (0,1)$ vorkommen extrahiert werden
- jeweils mit Angabe von Häufigkeit f
- in **einem** Durchlauf durch die Daten
 - also gut für Data Streams geeignet
- beweisbarer, möglichst kleiner Speicherbedarf

Was heißt näherungsweise ?

- dabei nicht exakt zählen, sondern mit $\epsilon \in (0,1)$ einstellbarer (garantierter) Genauigkeit
 - alle Elemente die häufiger als sN vorkommen werden gefunden
 - kein Element das seltener als $(s - \epsilon)N$ vorkommt wird ausgegeben
 - angegebene Raten sind kleiner gleich den richtigen um maximal ϵN



Algorithmen Grundgerüst

- Initialisierung $K = \emptyset$
- FOR_EACH neues eintreffendes Element e
 - IF bereits ein Eintrag in K für e existiert?
 - THEN in Datenstruktur Zähler für e inkrementieren
 - ELSE evtl. neuen Eintrag in K einfügen
 - IF „Zeit zum Aufräumen“ THEN
 - Elemente aus K entfernen
- **Ausgabe:** alle Elemente aus K , die häufig genug vorkamen

Sticky Sampling - Übersicht

- Stochastik bzw. Stichproben basiert
- Anforderungen nur mit wählbarer Wahrscheinlichkeit $1 - \delta$ erfüllt
- Speicher maximal $\frac{2}{\epsilon} \log(s^{-1} \delta^{-1})$

Sticky Sampling - Benennung

- Datenstruktur S ist Menge von Einträgen der Form (e, f)
 - e Element im Datenstrom
 - $f \in \mathbb{N}$ genäherte Häufigkeit
- Variable r für „sampling rate“ wird mitgeführt

Sticky Sampling - Algorithmus (1)

- Initialisierung $S := \emptyset$, $r := 1$, $t := \frac{1}{\epsilon} \log(s^{-1} \delta^{-1})$
- FOR_EACH neues eintreffendes Element e
 - IF bereits ein Eintrag in S für e existiert?
 - THEN entsprechendes f um eins erhöhen
 - ELSE Mit Wahrscheinlichkeit $\frac{1}{r}$ neuen Eintrag der Form $(e, 1)$ in S einfügen
 - IF $N = 2 \cdot r \cdot t$ THEN
 - $r := 2 \cdot r$
 - S ausdünnen (siehe nächste Folie)



Sticky Sampling - Algorithmus (2)

- S ausdünnen:
 - FOR_EACH $(e, f) \in S$
 - WHILE Münzwurf = „Kopf“
 - $f := f - 1$
 - IF $f = 0$ THEN lösche Element aus S
- **Ausgabe:** Alle Elemente $(e, f) \in S$ mit $f \geq (s - \epsilon) N$

Sticky Sampling - Beispiel

$$s=0,1 \quad \epsilon=0,01 \quad \delta=0,01 \quad t = \frac{1}{0,001} \log_2 \left(\frac{1}{0,1} \cdot \frac{1}{0,01} \right) = 1000$$

N=4000

| | | b | g | a | c | f | f | f | f | schnitt | f |
|-----|-------|---|--------|---|---|--------|---|---|---|---------|--------|
| a | 700 | | | + | | | | | | 699 | |
| b | 1221 | + | | | | | | | | 1221 | |
| c | 80 | | | | + | | | | | 79 | |
| d | 2 | | | | | | | | | delete | |
| e | 1511 | | | | | | | | | 1510 | |
| (f) | | | | | | 1/2->F | 1 | + | + | delete | 1/4->F |
| (g) | | | 1/2->F | | | | | | | | |

r=2
r=4

1/4 ->f := 4-seitiger Würfel landet nicht auf der richtigen Seite

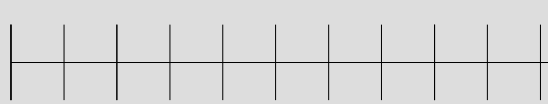
Lossy Counting - Übersicht

- deterministisch
- exakt in Anforderungsschranken
 - keinen Parameter δ
- Speicher maximal $\frac{1}{\epsilon} \log(\epsilon N)$
 - Im Gegensatz zu Sticky Sampling abhängig von N

Lossy Counting - Benennung

- Eingangsstrom in gedachte Behälter der Länge $w = \left\lceil \frac{1}{\epsilon} \right\rceil$ unterteilen
- nummeriere Behälter durch $id = 1, 2, \dots$
- aktuelle Behälternummer ist $id_{max} = \left\lceil \frac{N}{\epsilon} \right\rceil$
- Datenstruktur D Menge von Einträgen der Form (e, f, Δ)
 - e Element im Datenstrom
 - $f \in \mathbb{N}$ genäherte Häufigkeit
 - Δ maximal möglicher Fehler

Lossy Counting - Algorithmus

- Initialisierung $D = \emptyset$
- FOR_EACH neues eintreffendes Element e
 - IF bereits ein Eintrag in D für e existiert?
 - THEN entsprechendes f um eins erhöhen
 - ELSE neuen Eintrag der Form $(e, 1, id_{max} - 1)$ in D einfügen
 - IF $N \bmod w = 0$ THEN 
 - alle Elemente aus D löschen, für die gilt $f + \Delta \leq id_{max} = \frac{N}{w}$
- **Ausgabe:** alle Elemente $(e, f, \Delta) \in D$ mit $f \geq (s - \epsilon) N$

Lossy Counting - Beispiel

$$s=0,1 \quad \epsilon=0,01 \quad w = \left\lceil \frac{1}{\epsilon} \right\rceil = 100$$

N=4000 \rightarrow $id_{\max} = 100$

| | (f,Δ) | | b | g | a | c | f | f | f | schnitt | f | a |
|---|----------|-------|---|---|---|---|---|---|---|----------|---|---|
| a | (700,12) | | | | + | | | | | (701,12) | | + |
| b | (1221,1) | | + | | | | | | | (1222,1) | | |
| c | (80,20) | | | | | + | | | | (81,20) | | |
| d | (2,98) | | | | | | | | | -- | | |
| e | (1511,0) | | | | | | | | | (1511,0) | | |
| f | -- | | | | | | 1 | + | + | (3,98) | + | |
| g | -- | | | 1 | | | | | | -- | | |

Vergleich - Memory

Sticky Sampling

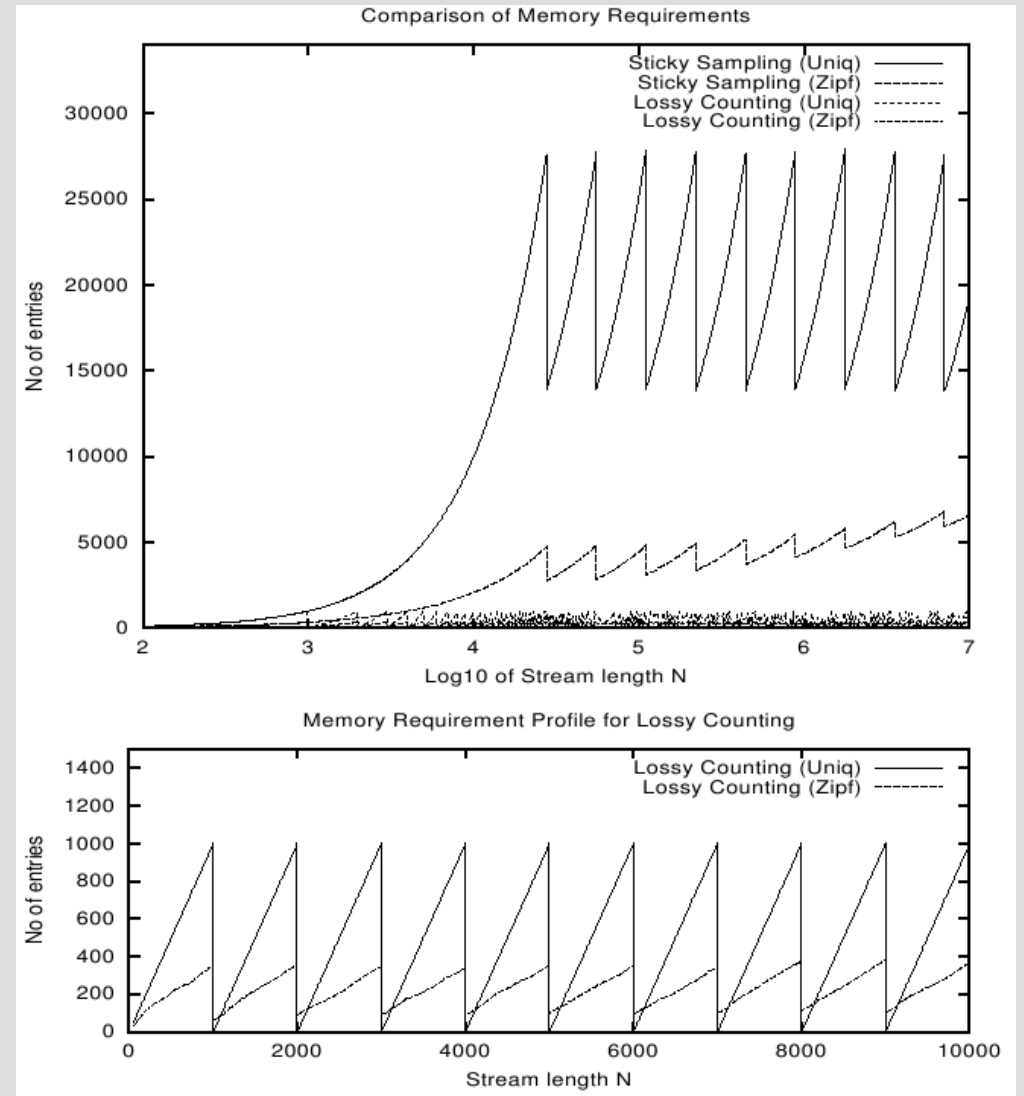
$$\frac{2}{\epsilon} \log(s^{-1} \delta^{-1})$$

Lossy Counting

$$\frac{1}{\epsilon} \log(\epsilon N)$$

- **Uniq:** Alle Elemente Einzigartig
- **Zipf:** Zipf-Verteilung mit Parameter 1,25
Häufigkeit proportional zu
1 / (Rangfolge der Elements in der
Gesamtmenge)

$$s = 10\% \quad \epsilon = 1\% \quad \delta = 0,1\%$$



Praxis

- Lossy Counting benötigt wesentlich weniger Speicher und ist praktisch nicht abhängig von N
- Beide Algorithmen arbeiten wesentlich genauer als die Angegebene Schranke es verlangt
 - Lossy Counting: Falls im ersten Fenster alle wichtigen Elemente vorkommen, ist die Zählung sehr wahrscheinlich exakt
- Lossy Counting prozessorlastig
 - geeignet um Daten direkt von Platte zu lesen ohne Performance Einbuße

Literatur

- J. Fürnkranz, G. Grieser. Vorlesungsskript „Maschinelles Lernen: Symbolische Ansätze“. WS2006/07
- G. S. Manku and R. Motwani. Approximate Frequency Counts over Streaming Data. In Proceedings of VLDB 2002, Aug. 2002
- Wikipedia. 11.11.2006 http://en.wikipedia.org/wiki/Zipf%27s_law
- Cordula Nimz. Seminarvortrag. 15.12.2005 <http://www.inf.fu-berlin.de/lehre/WS05/Seminar-Algorithmen/Vortrag20051215.pdf>