BJKST algorithm

Stream Programming Project Assignment

E. Chmielewski
M.J. Błaszczak

Problem – estimate a number of distinct elements

- It is a measure of data dispersion (“spread out”)
- Taking samples is misleading → tendency to pick frequent items

Fig. 1. Data stream, based on “What is data stream and how to use it”, www.onaudience.com

Model – vanilla streaming model

- A stream $\sigma = \langle a_{1,} a_{2,} \ldots, a_n \rangle, a_i \in [n]$ with frequency vectors $\boldsymbol{f} = (f_{1,} f_{2,} \ldots, f_n)$
- Number of distinct elements in the stream $d = |\{ j : f_j > 0 \}|$

- A task is to calculate $(\epsilon, \delta) - estimate$ for d where
  $d(\sigma) - output\ of\ a\ randomized\ streaming\ algorithm$
  $\phi - function\ that\ d\ calculates$
  and an algorithm $(\epsilon, \delta) - estimate\ \phi$ if $Pr\left[\left| \frac{d(\sigma)}{\phi(sigma)} \right| > \epsilon \right] \leqslant \delta$

[1] "Data Stream Algorithms Lecture Notes", Amit Chakrabarti

Algorithm 1\2 [The AMS99 “tidemark” algorithm]

$p - integer\,, p > 0$

$zeros(p) - number\ of\ zeros\ that\ the\ binary\ representation\ of\ p\ ends\ with$

$p - integer\,, p > 0$

$\textbf{Initialize}:$

$1: Choose\ a\ random\ hash\ function\ h:[n] \to [n] \ni 2 - universal\ hash\ family$

$2: z \leftarrow 0$

$\textbf{Process}(token\ j):$

$3: if\ zeros(h(j)) > z\ then\ z \leftarrow zeros(h(j))$

$\textbf{Output}: 2^{z+\frac{1}{2}}$

[1] “Data Stream Algorithms Lecture Notes”, Amit Chakrabarti

Algorithm 2\2 [The BJKST algorithm]

“leveraged” version of the AMS99

Initialize :

$1: Choose\ a\ random\ hash\ function\ h:[n] \rightarrow [n] \ni 2-univ.\ hash\ family$

$2: Choose\ a\ random\ hash\ function\ g:[n] \rightarrow [b\,\epsilon^{-4} \log^2 n] \ni 2-univ.\ hash\ family$

$3: z \leftarrow 0$

$4: B \leftarrow \emptyset$

Process $(token\ j)$:

$5: if\ zeros(h(j)) \geqslant z\ then$

$6: \quad B \leftarrow B \cup \{g(j), zeros(h(j))\}$

$7: \quad while\ |B| \geqslant c/\epsilon^2\ do$

$8: \qquad z \leftarrow z+1 \quad \wedge \quad B \leftarrow B \setminus (\alpha, \beta) \ \ \forall \ \ \beta < z$

Output $: |B| 2^z$

[1] “Data Stream Algorithms Lecture Notes”, Amit Chakrabarti

Algorithm 2\2 [The BJKST algorithm]

- b, c – universal constants to adjust estimation guarantee
- Algorithm try to determine size of the bucket B which:
  - consists of all $token\ j,\ \ zeros(h(j)) \geqslant z$
  - expectedly $d/2^z\, tokens$ should fall into this bucket
  - therefore estimetd size of the stream is $|B| 2^z$

- Space complexity: $O(\log n + (1/\epsilon^2)(\log 1/\epsilon) + \log\log n)$
- Optimality: the algorithm is close to optimal in its space usage