Lecture 4: Intro to Streaming

Piotr Indyk
Basic Data Stream Model

- Single pass over the data: \(i_1, i_2, \ldots, i_n\)
  - Typically, we assume \(n\) is known
- Bounded storage (typically \(n^\alpha\) or \(\log^c n\))
  - Units of storage: bits, words, coordinates or „elements“
    (e.g., points, nodes/edges)
- Fast processing time per element
  - Randomness OK (in fact, almost always necessary)

8 2 1 9 1 9 2 4 6 3 9 4 2 3 4 2 3 8 5 2 5 6  ...
Counting Distinct Elements

- Stream elements: numbers from \( \{1...m\} \)
- Goal: estimate the number of distinct elements \( DE \) in the stream
  - Up to \( 1 \pm \epsilon \)
  - With probability \( 1 - P \)
- Simpler goal: for a given \( T > 0 \), provide an algorithm which, with probability \( 1 - P \):
  - Answers YES, if \( DE > (1 + \epsilon)T \)
  - Answers NO, if \( DE < (1 - \epsilon)T \)
- Run, in parallel, the algorithm with
  \[ T = 1, 1 + \epsilon, (1 + \epsilon)^2, ..., n \]
  - Total space multiplied by \( \log_{1+\epsilon} n \approx \log(n) / \epsilon \)

Lecture 4
Vector Interpretation

Stream: 8 2 1 9 1 9 2 4 4 9 4 2 5 4 2 5 8 5 2 5

Vector X:

1 2 3 4 5 6 7 8 9

- Initially, $x=0$
- Insertion of $i$ is interpreted as $x_i = x_i + 1$
- Want to estimate $DE(x)$
Estimating \(\text{DE}(x)\) (based on [Flajolet-Martin’85])

- Choose a random set \(S\) of coordinates
  - For each \(i\), we have \(\Pr[i \in S] = 1/T\)
- Maintain \(\text{Sum}_S(x) = \sum_{i \in S} x_i\)
- Estimation algorithm \(I\):
  - YES, if \(\text{Sum}_S(x) > 0\)
  - NO, if \(\text{Sum}_S(x) = 0\)
- Analysis:
  - \(\Pr = \Pr[\text{Sum}_S(x) = 0] = (1-1/T)^{\text{DE}}\)
  - For \(T\) large enough: \((1-1/T)^{\text{DE}} \approx e^{-\text{DE}/T}\)
  - Using calculus, for \(\varepsilon\) small enough:
    - If \(\text{DE} > (1+\varepsilon)T\), then \(\Pr \approx e^{-(1+\varepsilon)} < 1/e - \varepsilon/3\)
    - If \(\text{DE} < (1-\varepsilon)T\), then \(\Pr \approx e^{-(1-\varepsilon)} > 1/e + \varepsilon/3\)

Vector \(X\):

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}
\]

Set \(S\):

\[
+ \quad + \quad + \quad (T=4)
\]
Estimating DE(x) ctd.

- We have Algorithm I:
  - If $DE > (1+\varepsilon)T$, then $Pr < 1/e-\varepsilon/3$
  - if $DE < (1-\varepsilon)T$, then $Pr > 1/e+\varepsilon/3$

- Algorithm II:
  - Select sets $S_1 \ldots S_k$, $k=O(\log(1/P)/\varepsilon^2)$
  - Let $Z =$ number of $\text{Sum}_{S_j}(x)$ that are equal to 0
  - By Chernoff bound, with probability $>1-P$
    - If $DE > (1+\varepsilon)T$, then $Z < k/e$
    - if $DE < (1-\varepsilon)T$, then $Z > k/e$

- Total space:
  - Decision version: $O(\log (1/P)/\varepsilon^2)$ numbers in range 0...n
  - Estimation: $O(\log(n)/\varepsilon \cdot \log (1/P)/\varepsilon^2)$ numbers in range 0...n
    (the probability of error is $O(P \log(n)/\varepsilon)$)

- Better algorithms known:
  - Theory: $O(1/\varepsilon^2 + \log n)$ bits [Kane-Nelson-Woodruff’10]
  - Practice: need 128 bytes for all works of Shakespeare, $\varepsilon \approx 10\%$ [Durand-Flajolet’03]
Comments

• Implementing $S$:
  – Choose a hash function $h: \{1..m\} \rightarrow \{1..T\}$
  – Define $S=\{i: h(i)=1\}$
  – We have $i \in S$ iff $h(i)=1$

• Implementing $h$
  – In practice: to compute $h(i)$ do:
    • Seed=$i$
    • Return random()
  – In theory: use pseudorandom generators (see an example in the next algorithm)
More comments

Start

Vector X:

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9
\end{array}
\]

- The algorithm uses “linear sketches”
  \[
  \text{Sum}_{S_j}(x) = \sum_{i \in S_j} x_i
  \]
- Can implement decrements \(x_i = x_i - 1\)
  - I.e., the stream can contain deletions of elements (as long as \(x \geq 0\))
  - Other names: dynamic model, turnstile model
More General Problem

- What other functions of a vector \( x \) can we maintain in small space?
- \( L_p \) norms:
  \[
  \|x\|_p = \left( \sum_i |x_i|^p \right)^{1/p}
  \]
  - We also have \( \|x\|_\infty = \max_i |x_i| \)
  - ... and we can define \( \|x\|_0 = \text{DE}(x) \), since \( \|x\|_p^p = \sum_i |x_i|^p \rightarrow \text{DE}(x) \) as \( p \rightarrow 0 \)
- Alternatively: frequency moments \( F_p = p\)-th power of \( L_p \) norms (exception: \( F_0 = L_0 \))
- How much space do you need to estimate \( \|x\|_p \) (for const. \( \varepsilon \))?
- Theorem:
  - For \( p \in [0,2] \): polylog \( n \) space suffices
  - For \( p > 2 \): \( n^{1-2/p} \) polylog \( n \) space suffices and is necessary

[Alon-Matias-Szegedy’96, Feigenbaum-Kannan-Strauss-Viswanathan’99, Indyk’00, Coppersmith-Kumar’04, Ganguly’04, Bar-Yossef-Jayram-Kumar-Sivakumar’02’03, Saks-Sun’03, Indyk-Woodruff’05]

Lecture 4
Estimating $L_2$ norm: AMS
Choose $r_1 \ldots r_m$ to be i.i.d. r.v., with $\Pr[r_i=1]=\Pr[r_i=-1]=1/2$

Maintain $Z=\sum_i r_i x_i$ under increments/decrements to $x_i$

Algorithm I:

$Y=Z^2$

“Claim”: $Y$ “approximates” $||x||_2^2$ with “good” probability
Analysis

• We will use Chebyshev inequality
  – Need expectation, variance

• The expectation of $Z^2 = (\sum_i r_i x_i)^2$ is equal to
  
  $$E[Z^2] = E[\sum_{i,j} r_i x_i r_j x_j] = \sum_{i,j} x_i x_j E[r_i r_j]$$

• We have
  – For $i \neq j$, $E[r_i r_j] = E[r_i] E[r_j] = 0$ – term disappears
  – For $i = j$, $E[r_i r_j] = 1$

• Therefore

  $$E[Z^2] = \sum_i x_i^2 = ||x||_2^2$$

  (unbiased estimator)
Analysis, ctd.

- The second moment of $Z^2 = (\sum r_i x_i)^2$ is equal to the expectation of $Z^4 = (\sum r_i x_i)(\sum r_i x_i)(\sum r_i x_i)(\sum r_i x_i)$
- This can be decomposed into a sum of:
  - $\sum (r_i x_i)^4 \rightarrow$ expectation $= \sum x_i^4$
  - $6 \sum_{i<j} (r_i r_j x_i x_j)^2 \rightarrow$ expectation $= 6 \sum_{i<j} x_i^2 x_j^2$
  - Terms involving single multiplier $r_i x_i$ (e.g., $r_1 x_1 r_2 x_2 r_3 x_3 r_4 x_4$) $\rightarrow$ expectation $= 0$

Total: $\sum x_i^4 + 6 \sum_{i<j} x_i^2 x_j^2$

- The variance of $Z^2$ is equal to
  \[
  E[Z^4] - E^2[Z^2] = \sum x_i^4 + 6 \sum_{i<j} x_i^2 x_j^2 - (\sum x_i^2)^2 \\
  = \sum x_i^4 + 6 \sum_{i<j} x_i^2 x_j^2 - \sum x_i^4 - 2 \sum_{i<j} x_i^2 x_j^2 \\
  = 4 \sum_{i<j} x_i^2 x_j^2 \\
  \leq 2 (\sum x_i^2)^2
  \]
Analysis, ctd.

• We have an estimator $Y = Z^2$
  – $E[Y] = \sum_i x_i^2$
  – $\sigma^2 = \text{Var}[Y] \leq 2 \left( \sum_i x_i^2 \right)^2$

• Chebyshev inequality:
  $$\Pr[|E[Y] - Y| \geq c\sigma] \leq \frac{1}{c^2}$$

• Algorithm II:
  – Maintain $Z_1 \ldots Z_k$ (and thus $Y_1 \ldots Y_k$), define $Y' = \sum_i Y_i / k$
  – $E[Y'] = k \frac{\sum_i x_i^2}{k} = \sum_i x_i^2$
  – $\sigma'^2 = \text{Var}[Y'] \leq 2k(\sum_i x_i^2)^2 / k^2 = 2(\sum_i x_i^2)^2 / k$

• Guarantee:
  $$\Pr[|Y' - \sum_i x_i^2| \geq c (2/k)^{1/2} \sum_i x_i^2] \leq 1/c^2$$

• Setting $c$ to a constant and $k = O(1/\varepsilon^2)$ gives $(1 \pm \varepsilon)$-approximation with const. probability
Comments

• Only needed 4-wise independence of $r_1 \ldots r_m$
  – I.e., for any subset $S$ of $\{1 \ldots m\}$, $|S|=4$, and any $b \in \{-1,1\}^4$, we have $\Pr[r_S=b]=1/2^4$
  – Can generate such vars from $O(\log m)$ random bits

• What we did:
  – Maintain a “linear sketch” vector $Z=[Z_1 \ldots Z_k] = R \cdot x$
  – Estimator for $||x||_2^2 : \frac{(Z_1^2 + \ldots + Z_k^2)}{k} = \frac{||Rx||_2^2}{k}$
  – “Dimensionality reduction”: $x \rightarrow Rx$
    … but the tail somewhat “heavy”
  – Reason: only used second moment of the estimator