Streaming Algorithms, etc.

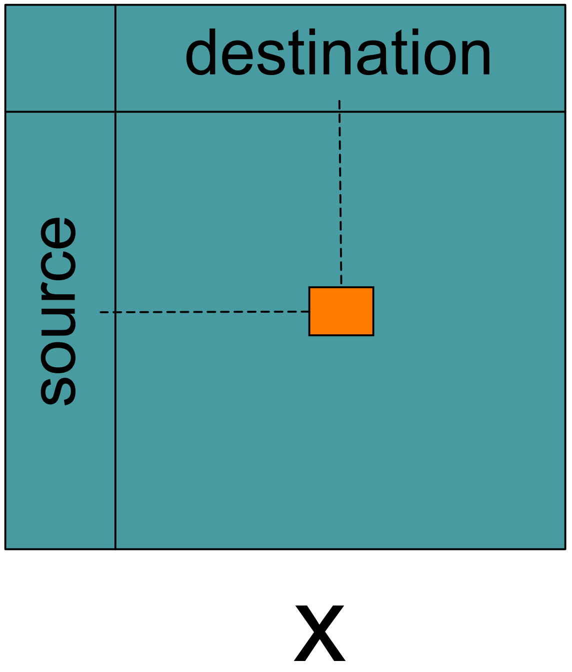
MIT
Piotr Indyk

Data Streams

- A data stream is a sequence of data that is too large to be stored in available memory (disk, memory, cache, etc.)
- Examples:
  - Network traffic
  - Database transactions
  - Sensor networks
  - Satellite data feed

Example application: Monitoring Network Traffic

- Router routs packets
  (many packets)
  - Where do they come from ?
  - Where do they go to ?
- Ideally, would like to maintain a traffic matrix x[.,.]
  - For each (src,dst) packet, increment $x_{src,dst}$
  - Requires way too much space!
    (2^{32} x 2^{32} entries)
  - Need to maintain a compressed version of the matrix



Data Streams

- A data stream is a (massive) sequence of data
  - Too large to store (on disk, memory, cache, etc.)
- Examples:
  - Network traffic (source/destination)
  - Database transactions
  - Sensor networks
  - Satellite data feed
  - …
- Approaches:
  - Ignore it
  - Develop algorithms for dealing with such data

This course

- Systematic introduction to the area
  - Emphasis on common themes
  - Connections between streaming, sketching, compressed sensing, communication complexity, …
  - ~~First~~ Second of its kind

    (previous edition from Fall'07: see my web page at MIT)

- Style: algorithmic/theoretical…
  - Background in linear algebra and probability

Topics

- Streaming model. Estimating distinct elements (L0 norm)
- Estimating L2 norm (AMS), Johnson Lindenstrauss
- Lp norm (p<2), other norms, entropy
- Heavy hitters: L1 norm, L2 norm, sparse approximations
- Sparse recovery via LP decoding
- Lower bounds: communication complexity, indexing, L2 norm
- Options: MST, bi-chromatic matching, insertions-only streams, Fourier sampling,

Plan For This Lecture

- Introduce the data stream model(s)
- Basic algorithms
  - Estimating number of distinct elements in a stream
  - Into to frequency moments and norms

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^α or $\log^c n$)
  - Units of storage: bits, words 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 $1\pm\varepsilon$
  - With probability $1-P$
- Simpler goal: for a given $T>0$, provide an algorithm which, with probability $1-P$:
  - Answers YES, if $DE > (1+\varepsilon)T$
  - Answers NO, if $DE < (1-\varepsilon)T$
- Run, in parallel, the algorithm with

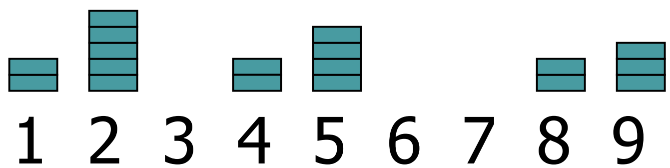
  $$T=1,\ 1+\varepsilon,\ (1+\varepsilon)^2, \ldots,\ n$$

  - Total space multiplied by $\log_{1+\varepsilon} n \approx \log(n)/\varepsilon$
  - Probability of failure multiplied by the same factor

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:



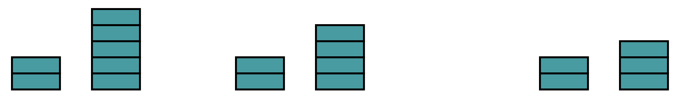
- Initially, $x=0$
- Insertion of i is interpreted as

$$x_i = x_i + 1$$

- Want to estimate $DE(x) = ||x||_0$

Estimating DE(x)

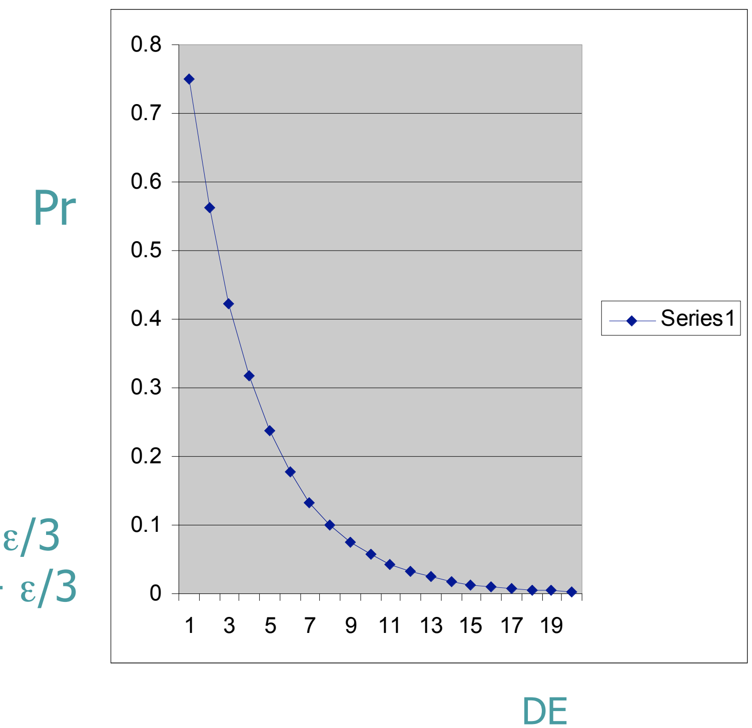
Vector X:



1 2 3 4 5 6 7 8 9

Set S: + + + (T=4)

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



Estimating DE(x) ctd.

- We have Algorithm A:
  - If $DE > (1+\varepsilon)T$, then $Pr < 1/e - \varepsilon/3$
  - if $DE < (1-\varepsilon)T$, then $Pr > 1/e + \varepsilon/3$
- Algorithm B:
  - Select sets $S_1 \ldots S_k$, $k = O(\log(1/P)/\varepsilon^2)$
  - Let Z = number of $Sum_{Sj}(x)$ that are equal to 0
  - By Chernoff bound (define), with probability $>1-P$
    - If $DE > (1+\varepsilon)T$, then $Z < k/e$
    - if $DE < (1-\varepsilon)T$, then $Z > k/e$

- Total space: $O(\log(n)/\varepsilon \;\; \log(1/P)/\varepsilon^2)$ numbers in range $0 \ldots n$
- Can remove the $\log(n)/\varepsilon$ factor
- Bibliographic note: [Flajolet-Martin'85]

Interlude – Chernoff bound

- Let $Z_1 \ldots Z_k$ be i.i.d. Bernoulli variables, with $\Pr[Z_j=1]=p$
- Let $Z=\sum_j Z_j$
- For any $1>\epsilon>0$, we have

  $\Pr[\ |E[Z]-Z| > \varepsilon E[Z]\ ] \leq 2\exp(\ -\varepsilon^2 E[Z]/3\ )$

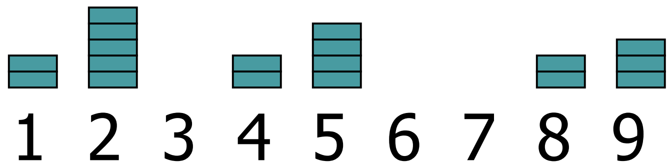
Comments

- Implementing S:
  - Choose a hash function h: {1..m} -> {1..T}
  - Define S={i: h(i)=1}
- Implementing h
  - Pseudorandom generators. More later.
- Better algorithms known:
  - Theory: $O(\log(1/\varepsilon)/\varepsilon^2 + \log n)$ bits

    [Bar-Yossef-Jayram-Kumar-Sivakumar-Trevisan'02]
  - Practice: need 128 bytes for all works of Shakespeare , $\varepsilon \approx 10\%$ [Durand-Flajolet'03]

More comments

Vector X:



- The algorithm uses "linear sketches"

$$Sum_{Sj}(x)=\Sigma_{i\in Sj}\ x_i$$

- 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:

$$||x||_p = (\sum_i |x_i|^p)^{1/p}$$

  - We also have $||x||_\infty = \max_i |x_i|$
  - ... and $||x||_0 = DE(x)$, since $||x||_p^p = \sum_i |x_i|^p \to DE(x)$ as $p \to 0$
- 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. ε) ?
- 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]