Insertions-only streams

Piotr Indyk
MIT
The “great streaming divide”

- Insertions and deletions
  - Linear mappings
  - Randomized algorithms (typically)
  - Covered so far
- Insertions-only
  - Divide and conquer, etc.
  - Deterministic algorithms (typically)
  - Will cover in the next few lectures
Today

• L1 heavy hitters’ using $O(1/\varphi)$ words of space
  – Compare to $O(1/\varphi \log m)$ if deletions allowed
• $O_\alpha(1)$-Approximate Metric k-median using $O(kn^\alpha)$ space, for any $\alpha > 0$

$n$ is the length of the stream
Heavy Hitters

• Define (today)
  \[ HH_\phi (x) = \{ i: |x_i| > \phi \ |x|_1 = \phi n \} \]

• Heavy Hitters’ Problem:
  – Parameter: \( \phi \)
  – Goal: return a set \( S \) of coordinates s.t.
    • \( S \) contains \( HH_\phi (x) \)
    • \( S \) has size \( O(1/\phi) \)

• The basic idea by [Misra-Gries’82]
  – Presentation from [Demaine-Munro-OrtizLopez’02]
Warmup: $\phi = \frac{1}{2}$

• If there exists a “majority” element, we want to find it

• One-counter algorithm:
  – Set $c=0$
  – For each stream element $a$
    • If $c=0$ then $e=a$
    • If $e=a$ then $c=c+1$ else $c=c-1$
  – Report $e$
Correctness

• Assume majority element (MAJ) exists
• Algorithm recap:
  – Set $c=0$
  – For each stream element $a$
    • (invariant)
    • If $c=0$ then $e=a$
    • If $e = a$ then $c=c+1$ else $c=c-1$
  – Report $e$
• Invariant: if $c=0$ then MAJ is the majority element in the remainder of the stream (including $a$)
  – Between the times when $c=0$, the stream contains exactly 50% of $e$’s
General $\varphi = 1/(L+1)$

- **Algorithm:**
  - Set $S = \emptyset$; $c: S \rightarrow \{1 \ldots n\}$
  - For each stream element $a$
    - If $a \notin S$ and $|S| < L$ then add $a$ to $S$ and set $c(a) = 0$
    - If $a \in S$ then $c(a) = c(a) + 1$ else $c(a') = c(a') - 1$ for all $a' \in S$
    - For all $a' \in S$, if $c(a') = 0$ then remove $a'$ from $S$
  - Report $S$

- **Main proof idea:** each time we decrement all counters in $S$ (the “decrement event”) can be charged to $L+1$ unique stream elements
Correctness

• **Iteration:**
  – If \( a \not \in S \) and \(|S| < L \) then add \( a \) to \( S \) and set \( c(a) = 0 \)
  – If \( a \in S \) then \( c(a) = c(a) + 1 \) else \( c(a') = c(a') - 1 \) for all \( a' \in S \)
  – For all \( a' \in S \), if \( c(a') = 0 \) then remove \( a' \) from \( S \)

• **Proof:**
  – Consider any element \( a \) which occurs \( > \frac{n}{L+1} \) times
  – Denote:
    • \( t_f \) = number of decrement events while reading \( a \not \in S \)
    • \( t_d \) = number of decrement events while \( a \in S \)
    • \( t_i \) = number of increments while reading \( a \in S \)
    • \( t = t_f + t_i \) = total number of occurrences of \( a \)
  – By the previous slide observation, we have \( (L+1)(t_f + t_d) \leq n \)
  – If the final count of \( a \) is zero, then \( t_d = t_i \), and therefore
    \[ \frac{n}{L+1} < t = t_f + t_i = t_f + t_d \leq \frac{n}{L+1} \]
    which is a contradiction
Stronger guarantees

• One can show that
  \[ |c(a)-x_a| \leq n/(L+1) \]

• In fact, even
  \[ |c(a)-x_a| \leq \text{Err}_1^k(x) \]
Metric k-median

- Oracle access to a metric function $D(.,.)$
- Stream: a sequence of metric points $p$ defining a set $S$, $|S|=n$
- Definitions:
  - $T(p) = \min_{t \in T} D(p, t)$
  - For $|C|=k$, $\text{cost}(S, C) = \sum_{p \in S} C(p)$
  - $\text{cost}(S, Q) = \min_{C \subseteq Q, |C|=k} \text{cost}(S, C)$
- Goal: approximate $\text{cost}(S, S)$ and report the medians
k-median algorithm

- We show an $O(1)$-approximate algorithm using space $O((nk)^{1/2})$
  - Recursive application gives $O(n^{\alpha}k)$, any $\alpha>0$
- Algorithm from [Guha-Mishra-Motwani-O’Callaghan’00]
- We will need an off-line $b$-approximate algorithm that uses linear space
  - [Arya-Garg-Khandekar-Meyerson-Munagala-Pandit’01] does the job, with $b=3+\varepsilon$
Algorithm

• Partition the stream into blocks $S_1 \ldots S_L$, $L=\frac{n}{(n/k)^{1/2}}$, each of size $\frac{n}{L} = \frac{n}{(nk)^{1/2}}$

• For each $S_i$
  - Find medians $\{c_1^i \ldots c_k^i\}$ which $b$-approximate $\text{cost}(S_i, S_i)$
  - Compute $w_{ij}^i = \text{number of points in } S_i \text{ assigned to } c_j^i$

• Find $b$-approximate $k$ medians $C'$ for the weighted set $W=\{w_{11}^1 c_1^1 \ldots w_{Lk}^L c_k^L\}$
  - $wc$ means points $c$ “copied” $w$ times

“Medians of weighted medians are medians”
Approximation

- Notation:
  - cost=$\text{cost}(S,S)$
  - $C$ = the optimum set of medians, i.e., $\text{cost}(S,C)=\text{cost}(S,S)$
- Fact 1: $\text{cost}(S,S) \leq 2\text{cost}(S,Q)$
  - Can replace each median by the closest point in $S$
- Fact 2: $\sum_i \text{cost}(S_i, S_i) \leq 2\text{cost}(S,C)$
- Proof:
  - From Fact 1, we have $\text{cost}(S_i, S_i) \leq 2\text{cost}(S_i, S)$
  - Therefore
    $\sum_i \text{cost}(S_i, S_i) \leq 2 \sum_i \text{cost}(S_i, S) \leq 2 \sum_i \text{cost}(S_i, C)=2\text{cost}(S,C)$
- Therefore, the algorithm will find $(nk)^{1/2}$ weighted medians $W$ with cost at most $2b\text{cost}$
- Now we just have to connect $W$ to just $k$ medians with cost $O(\text{cost})$
- We will connect them to $C$
Approximation, ctd.

- Fact 3: \( \text{cost}(W,C) \leq 2b\text{cost} + \text{cost} \)

- Proof:
  - Notation:
    - \( q \in W \) is a single point (possibly out of many duplicates)
    - \( p \in S \) that is assigned to \( q \)
    - \( c \in C \) is a median to whom \( p \) is assigned to
  - We can connect each \( q \) to \( c \) through \( p \)
  - Total cost:
    - \( q \) to \( p \): \( 2b\text{cost} \)
    - \( p \) to \( c \): \( \text{cost} \)
  - Therefore, \( \text{cost}(W,C) \leq 2b\text{cost} + \text{cost} \)
Approximation, ctd.

• Altogether:
  – \( \text{cost}(S,W) \leq 2b \times \text{cost} \)
  – \( \text{cost}(W,C') \leq b \times \text{cost}(W,W) \)
  
    \[ \leq 2b \times \text{cost}(W,C) \leq 2b \times (2b \times \text{cost} + \text{cost}) \]
  – The total cost is \((4b^2 + 2b) \times \text{cost}\)
k-median wrap up

- $O(1)$-Approximate Metric k-median using $O(kn^\alpha)$ space, for any constant $\alpha > 0$
- Can achieve $O(k \log^2 n)$
  
  [Charikar-O'Callaghan-Panigrahy’03], based on [Meyerson’01]