# Heavy Hitters

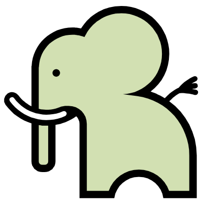
Piotr Indyk
MIT

Lecture 4

# Last Few Lectures

- Recap (last few lectures)
  - Update a vector x
  - Maintain a linear sketch
  - Can compute $L_p$ norm of x
    (in zillion different ways)
- Questions:
  - Can we do anything else ??
  - Can we do something about linear space bound for $L_\infty$ ??

# Heavy Hitters



- Also called frequent elements and elephants
- Define

$$HH^p_\varphi(x) = \{ i : |x_i| \geq \varphi \, ||x||_p \}$$

- $L_p$ Heavy Hitter Problem:
  - Parameters: $\varphi$ and $\varphi'$ (often $\varphi' = \varphi - \varepsilon$)
  - Goal: return a set $S$ of coordinates s.t.
    - $S$ contains $HH^p_\varphi(x)$
    - $S$ is included in $HH^p_{\varphi'}(x)$
- $L_p$ Point Query Problem:
  - Parameter: $\alpha$
  - Goal: at the end of the stream, given $i$, report

$$x^*_i = x_i \pm \alpha \, ||x||_p$$

# Which norm is better ?

- Since $||x||_1 \geq ||x||_2 \geq \ldots \geq ||x||_\infty$, we get that the higher Lp norms are better
- For example, for Zipfian distributions $x_i=1/i^\beta$, we have
  - $||x||_2$ : constant for $\beta>1/2$
  - $||x||_1$ : constant only for $\beta>1$
- However, estimating higher Lp norms tends to require higher dependence on $\alpha$

# A Few Facts

- Fact 1: The size of $HH^p_\varphi(x)$ is at most $1/\varphi$
- Fact 2: Given an algorithm for the $L_p$ point query problem, with:
  - parameter $\alpha$
  - probability of failure $<1/(2m)$

  one can obtain an algorithm for $L_p$ heavy hitters problem with:
  - parameters $\varphi$ and $\varphi' = \varphi-2\alpha$ (any $\varphi$)
  - same space (plus output)
  - probability of failure $<1/2$

  Proof:
  - Compute all $x_i^*$ (note: this takes time $O(m)$ )
  - Report $i$ such that $x_i^* \geq \varphi-\alpha$

# $L_2$ point query

# Point query

- We start from $L_2$
- A few observations:
  - $x_i = x * e_i$
  - For any $u, v$ we have

$$||u-v||^2 = ||u||^2 + ||v||^2 - 2u*v$$

- Algorithm [Gilbert-Kotidis-Muthukrishnan-Strauss'01]
  - Maintain a sketch $Rx$, with failure probability $P$
  - Assume $s = ||Rx||_2 = (1 \pm \varepsilon)||x||_2$
  - Estimator:

$$Y = ( 1 - || Rx/s - Re_i ||^2/2 ) s$$

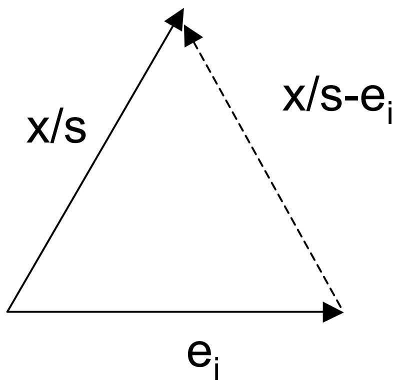
# Intuition

- Ignoring the sketching function R, we have

  $(1-||x/s-e_i||^2/2)s$

  $= (1-||x/s||^2/2 - ||e_i||^2/2 + x/s\ e_i)\ s$

  $= (1-1/2-1/2+x/s\ e_i)s = xe_i$




- Now we just need to deal with epsilons

# Analysis of $Y=(1-||Rx/s - Re_i||^2/2)s$

$|| Rx/s - Re_i ||^2/2$
$= || R(x/s-e_i) ||^2/2$
$= (1\pm\varepsilon)|| x/s - e_i ||^2/2$ Holds with prob. $1-P$
$= (1\pm\varepsilon)|| x/(||x||_2(1\pm\varepsilon)) - e_i ||^2/2$
$= (1\pm\varepsilon)[ 1/(1\pm\varepsilon)^2 + 1 - 2x^*e_i/( ||x||_2(1\pm\varepsilon))]/2$
$= (1\pm c\varepsilon)(1 - x^*e_i/||x||_2 )$

$Y$
$= [ 1 - (1\pm c\varepsilon)(1 - x^*e_i/||x||_2) ] \quad ||x||_2(1\pm\varepsilon)$
$= [ 1 - (1\pm c\varepsilon) + (1\pm c\varepsilon)x^*e_i/||x||_2 ] \quad ||x||_2(1\pm\varepsilon)$
$= [ \pm c\varepsilon ||x||_2 + (1\pm c\varepsilon)x^*e_i ] (1\pm\varepsilon)$
$= \pm c'\varepsilon ||x||_2 + x^*e_i$

# Altogether

- Can solve $L_2$ point query problem, with parameter $\alpha$ and failure probability $P$ by storing $O(1/\alpha^2 \log(1/P))$ numbers
- Pros:
  - General reduction to $L_2$ estimation
  - Intuitive approach (modulo epsilons)
  - In fact $e_i$ can be an arbitrary unit vector
- Cons:
  - Constants in the analysis are large
- There is a more direct approach using AMS sketches [A-Gibbons-M-S'99], with better constants

# $L_1$ Point Queries/Heavy Hitters

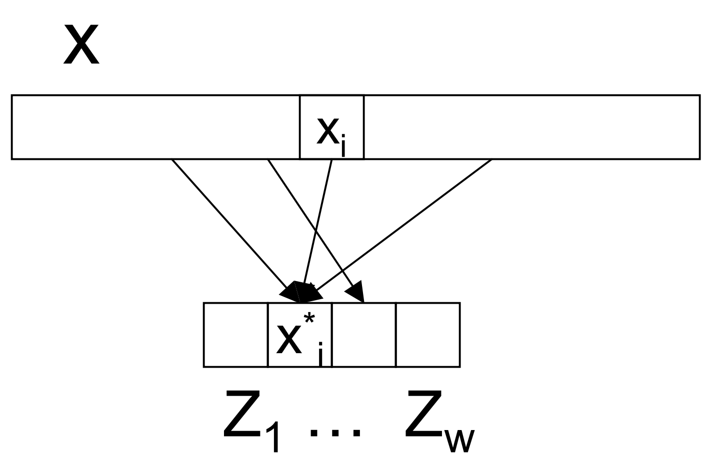
# $L_1$ Point Queries/Heavy Hitters

- For starters, assume $x \geq 0$

  (not crucial, but then the algorithm is really nice)
- Point queries: algorithm A:
  - Set $w=2/\alpha$
  - Prepare a random hash function $h: \{1..m\} \rightarrow \{1..w\}$
  - Maintain an array $Z=[Z_1, \ldots Z_w]$ such that

    $$Z_j = \sum_{i:\ h(i)=j} x_i$$
  - To estimate $x_i$ return

    $$x^*_i = Z_{h(i)}$$

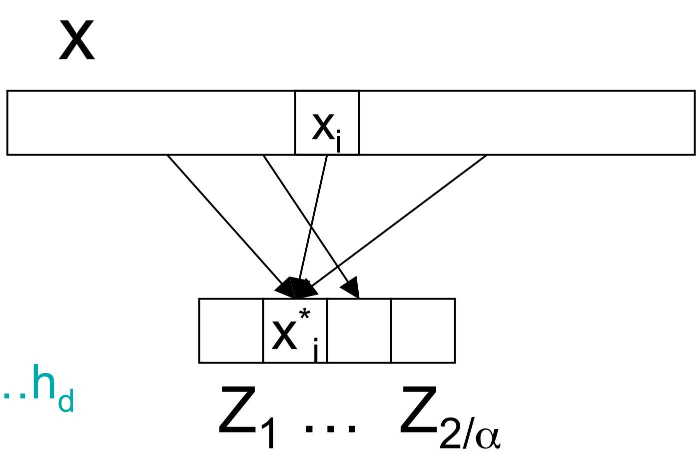


# Analysis

- Facts:
  - $x^*_i \geq x_i$
  - $E[\ x_i^* - x_i\ ] = \sum_{l \neq i} \Pr[h(l)=h(i)] x_l \leq \alpha/2\ ||x||_1$
  - $\Pr[\ |x_i^* - x_i| \geq \alpha\ ||x||_1\ ] \leq 1/2$
- Algorithm B:
  - Maintain $d$ vectors $Z^1 \ldots Z^d$ and functions $h_1 \ldots h_d$
  - Estimator:

$$x_i^* = \min_t Z^t_{ht(i)}$$

- Analysis:
  - $\Pr[\ |x_i^* - x_i| \geq \alpha\ ||x||_1\ ] \leq 1/2^d$
  - Setting $d=O(\log m)$ sufficient for $L_1$ Heavy Hitters
- Altogether, we use space $O(1/\alpha\ \log m)$
- For general $x$:
  - replace "min" by "median"
  - adjust parameters (by a constant)



# Comments

- Can reduce the recovery time to about O(log m)
- Other goodies as well
- For details, see

[Cormode-Muthukrishnan'04]: "The Count-Min Sketch…"

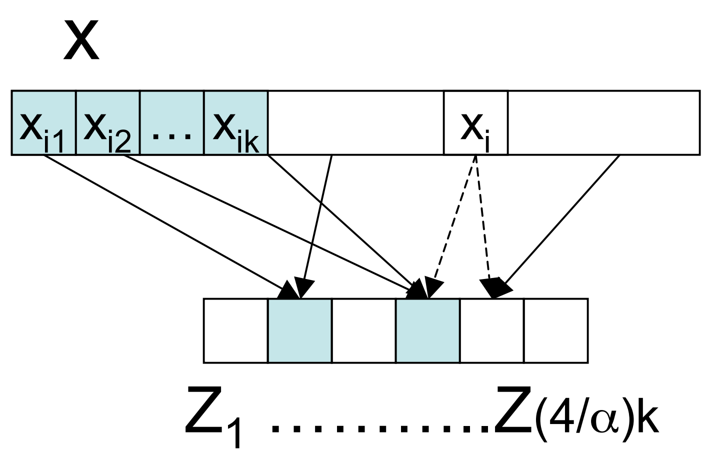
- Also:
  - [Charikar-Chen-FarachColton'02]
    (variant for the $L_2$ norm)
  - [Estan-Varghese'02]
  - Bloom filters

# Sparse Approximations

- Sparse approximations (w.r.t. $L_p$ norm):
  - For a vector x, find x' such that
    - x' has "complexity" k
    - $||x-x'||_p \le (1+\alpha)$ Err , where $Err=Err^p_k = \min_{x"} ||x-x"||_p$,
      for x" ranging over all vectors with "complexity" k
  - Sparsity (i.e., $L_0$ ) is a very natural measure of complexity
    - In this case, best x' consists of k coordinates of x that are largest in magnitude, i.e., "heavy hitters"
    - Then the error is the $L_p$ norm of the "non-heavy hitters", a.k.a. "mice"
- Question: can we modify the previous algorithm to solve the sparse approximation problem ?
- Answer: YES
  [Charikar-Chen-FarachColton'02, Cormode-Muthukrishnan'05] (for $L_2$ norm))
- Just set $w=(4/\alpha)k$
- We will see it for the $L_1$ norm

# Point Query

- We show how to get an estimate
  $x_i^* = x_i \pm \alpha\ Err/k$
- Assume
  $|x_{i1}| \geq \ldots \geq |x_{im}|$
- $\Pr[\ |x^*_i - x_i| \geq \alpha\ Err/k]$ is at most
  $\Pr[\ h(i) \in h(\{i1..ik\})\ ]$
  $+\quad \Pr[\ \sum_{l>k:\ h(il)=h(i)} x_l \geq \alpha\ Err/k\ ]$
  $\leq\quad 1/(2/\alpha) + 1/4$
  $<\quad 1/2$ (if $\alpha<1/2$)
- Applying min/median to $d=O(\log m)$ copies of the algorithm ensures that w.h.p
  $|x^*_i - x_i| < \alpha Err/k$



# Sparse Approximations

- Algorithm:
  – Return a vector $x'$ consisting of largest (in magnitude) elements of $x^*$
- Analysis (new proof)
  – Let $S$ (or $S^*$ ) be the set of $k$ largest in magnitude coordinates of $x$ (or $x^*$ )
  – Note that $||x^*_S|| \le ||x^*_{S^*}||_1$
  – We have

$$
\begin{aligned}
||x-x'||_1 &\le ||x||_1 - ||x_{S^*}||_1 + ||x_{S^*}-x^*_{S^*}||_1 \\
&\le ||x||_1 - ||x^*_{S^*}||_1 + 2||x_{S^*}-x^*_{S^*}||_1 \\
&\le ||x||_1 - ||x^*_S||_1 + 2||x_{S^*}-x^*_{S^*}||_1 \\
&\le ||x||_1 - ||x_S||_1 + ||x^*_S-x_S||_1 + 2||x_{S^*}-x^*_{S^*}||_1 \\
&\le Err + 3\alpha/k * k *Err \\
&\le (1+3\alpha)Err
\end{aligned}
$$

# Altogether

- Can compute $k$-sparse approximation to $x$ with error $(1+\alpha)Err^1_k$ using $O(k/\alpha \log m)$ space (numbers)
- This also gives an estimate

$$x_i^* = x_i \pm \alpha\, Err^1_k/k$$