Lecture 1. The data stream model. Counting. Probability tools

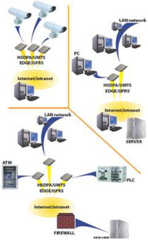
Ricard Gavaldà

MIRI Seminar on Data Streams, Spring 2015

Contents

Data streams everywhere

Data streams everywhere



- Telcos - phone calls
- Satellite, radar, sensor data
- Computer systems and network monitoring
- Search logs, access logs
- RSS feeds, social network activity
- Websites, clickstreams, query streams
- E-commerce, credit card sales
- ...

Example 1: Online shop

Thousands of visits / day

- Is this “customer” a robot?
- Does this customer want to buy?
- Is customer lost? Finding what s/he wants?
- What products should we recommend to this user?
- What ads should we show to this user?
- Should we get more machines from the cloud to handle incoming traffic?

Example 2: Web searchers

Millions of queries / day

- What are the top queries right now?
- Which terms are gaining popularity now?
- What ads should we show for this query and user?

Example 3: Phone company

Hundreds of millions of calls/day

- Each call about 1000 bytes per switch
- I.e., about 1Tb/month; must keep for billing

- Is this call fraudulent?
- Why do we get so many call drops in area X?
- Should we reroute differently tomorrow?
- Is this customer thinking of leaving us?
- How to cross-sell / up-sell this customer?

Example 4: Network link

Several Gb /minute at UPC's outlink
Really impossible to store

- Detect abusive users
- Detect anomalous traffic patterns
- . . . DDOS attacks, intrusions, etc.

Others

- Social networks: Planet-scale streams
- Smart cities. Smart vehicles
- Internet of Things
- (more phones connected to devices than used by humans)
- Open data; governmental and scientific

- We generate far more data than we can store

Data Streams: Modern times data

- Data arrives as sequence of items
- At high speed
- Forever
- Can't store them all
- Can't go back; or too slow
- Evolving, non-stationary reality

In algorithmic words. . .

The Data Stream axioms:

1. One pass
2. Low time per item - read, process, discard
3. Sublinear memory - only summaries or sketches
4. Anytime, real-time answers
5. The stream evolves over time

Course outline

Part I:

- The data stream model. Probability tools
- Statistics on streams; frequent elements
- Sketches for linear algebra and graphs
- Dealing with change

Part II:

- Predictive models
- Evaluation
- Clustering
- Frequent pattern mining
- Distributed stream mining

The data stream model

Computing in data streams

- Approximate answers are often OK
- Specifically, in learning and mining contexts
- Often computable with surprisingly low memory, one pass

Main Ingredients: Approximation and Randomization

- Algorithms use a source of independent random bits
- So different runs give different outputs
- But “most runs” are “approximately correct”

Randomized Algorithms

(ε, δ)-approximation

A randomized algorithm A (ε, δ)-approximates a function $f : X \to R$ iff for every $x \in X$, with probability $\geq 1 - \delta$

- (absolute approximation) $|A(x) - f(x)| < \varepsilon$
- (relative approximation) $|A(x) - f(x)| < \varepsilon f(x)$

Often ε, δ given as inputs to A
ε = accuracy; δ = confidence

Notation

$a \simeq b$ means "= up to lower order terms", $a \simeq a(1 + o(1))$

$a \sim b$ means whatever I find convenient at that point

log is base 2 unless otherwise noted

$\widetilde{O}(.)$ hides "polylog" terms, e.g. $\sqrt{n}\log^3 n \in \widetilde{O}(\sqrt{n})$

Three problems on Data Streams

Four examples:

- Counting distinct elements
- Finding heavy hitters
- Counting in a sliding window

Counting distinct elements

- How many distinct IP addresses has the router seen?
- An IP may have passed once, or many many times

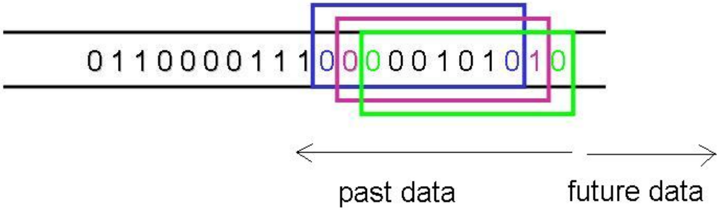
- **Fact:** Any algorithm must use $\Omega(n)$ memory to solve this problem exactly on a data stream, where n is number of different IPs seen
- **Fact:** $O(\log n)$ suffices to approximate within 1%

Finding heavy hitters

- Which IP's have used over ε fraction of bandwidth (each)? (Note: There can't be more than $1/\varepsilon$ of these)

- **Fact:** Any algorithm must use $\Omega(n)$ memory to solve this problem exactly on a data stream, where n is number of distinct IPs seen
- **Fact:** $O(1/\varepsilon)$ memory suffices if we allow a constant error factor

Counts in a sliding window



- Stream of bits; fixed n
- Question: "how many 1's were there among the last n"?
- **Fact:** Any algorithm must use $\Omega(n)$ memory to solve this problem exactly on a data stream
- **Fact:** $O(\log n)$ suffices to approximate within 1%

My main argument for sketches

If we keep one count, it's ok to use a lot of memory

If we have to keep many counts, they should use low memory

When learning / mining, we need to keep many counts

$\therefore$ Sketching is a good basis for data stream learning / mining

Approximate Counting

Counting

Most basic question?

How many items have we read so far in the data stream?

To count up to t elements *exactly*, $\log t$ bits are *necessary*

Next is an approximate solution using $\log \log t$ bits

Approximate counting

Saving k bits

Init: $c \leftarrow 0$;

Update:

draw a random number $x \in [0, 1]$;
if $(x \leq 2^{-k})$ $c{+}{+}$;

Query: return $2^k c$;

$$E[c] = t/2^k, \qquad \sigma[c] \simeq \sqrt{t/2^k}$$

Space $\log t - k \rightarrow$ we saved k bits!

Morris' approximate counter [Morris 77]

Morris' counter

Init: $c \leftarrow 0$;

Update:

  draw a random number $x \in [0,1]$;
  if $(x \leq 2^{-c})$ $c{+}{+}$;

Query: return $2^c - 2$;

$E[c] \simeq \log t$
$E[2^c] = t + 2$
$\sigma[2^c] \simeq t/\sqrt{2} \simeq 0.7\,t$

Morris' approximate counter

- Memory = memory used to keep c = $\log c = \log\log t$
- Can count up to 1 billion with $\log\log 10^9 = 5$ bits
- Problem: large variance, $O(t)$

Reducing the variance, method I

Use basis $b < 2$ instead of basis 2:

- Places t in the series $1, b, b^2, \ldots, b^i, \ldots$ ("resolution" b)
- $E[b^c] \simeq t$, $\sigma[b^c] \simeq t \cdot \sqrt{(b-1)/2}$
- Space $\log\log t - \log\log b \quad (> \log\log t$, because $b < 2)$
- For $b = 1.2$, 20% of original variance, 2 extra bits

Reducing the variance, method II

- Run r parallel, independent copies of the algorithm
- On Query, average their estimates
- $E[Q] \simeq t$, $\sigma[Q] \simeq t/\sqrt{2r}$ (why?)
- Space $r \log\log t$
- Time per item multiplied by r

Worse performance, but more generic technique

Morris' counter: A non-streaming application

In [VanDurme+09]

- Counting k-grams in a large text corpus
- Number of k-grams grows exponentially with k
- Highly diverse frequencies
- Use Morris' counters (5 bits) instead of standard counters

Morris' counter: An improvement?

Exercise 1

Suppose in the Morris' counter I change

$$\text{if } (x \leq 2^{-c})\ c{+}{+};$$

to

$$\text{if } (x \leq 2^{-2^c})\ c{+}{+};$$

I claim this gives an algorithm using $\log\log\log t$ bits between updates (plus temporary $\log\log t$ memory during an update)

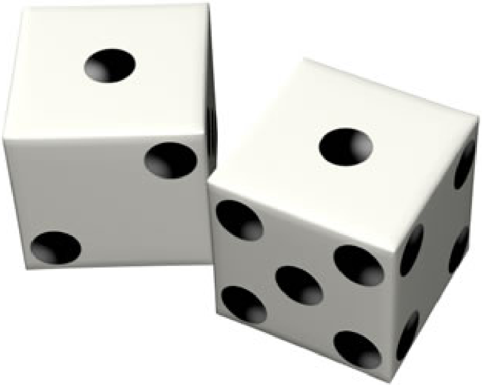
1) do you believe this?

2) if you do, think why this algorithm is not very useful, anyway

Hint: resolution

Probability and Sampling

Probability and Sampling



Probabilities

- A, B events
- $\Pr[A|B] = \Pr[A \wedge B] / \Pr[B]$
- A and B independent iff $\Pr[A \wedge B] = \Pr[A] \cdot \Pr[B]$
- equivalently, iff $\Pr[A|B] = \Pr[A]$
- Union bound:
  $\Pr[A \vee B] = \Pr[A] + \Pr[B] - \Pr[A \wedge B] \leq \Pr[A] + \Pr[B]$
- More in general, $\Pr[\bigvee_{i \in I} A_i] \leq \sum_{i \in I} \Pr[A_i]$

Expectation

(Discrete distributions)

- X real-valued random variable
- Expectation of X = $E[X] = \sum_x \Pr[X = x] \cdot x$
- $E[X - E[X]] = 0$
- Linearity of expectation:

  $E[X + Y] = E[X] + E[Y], \quad E[\alpha \cdot X] = \alpha \cdot E[X]$
- More in general, $E[\sum_{i \in I} \alpha_i \cdot X_i] = \sum_{i \in I} \alpha_i \cdot E[X_i]$
- If X and Y independent, $E[X \cdot Y] = E[X] \cdot E[Y]$

Variance

- Variance: $Var(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$
- $Var(\alpha \cdot X + \beta) = \alpha^2 \cdot Var(X)$
- If X and Y independent, $Var(X+Y) = Var(X) + Var(Y)$
- In general, if X_i are all independent and $Var(X_i) = \sigma^2$,

$$Var(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n^2}(n\sigma^2) = \frac{\sigma^2}{n}$$

  Equivalently,

$$\sigma(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{\sigma}{\sqrt{n}}.$$

Deviation Bounds

Markov's inequality

For a non-negative random variable X and every k

$$\Pr[X \geq k\,E[X]] \leq 1/k$$

Proof:

$$\begin{aligned} E[X] &= \sum_x \Pr[X = x] \cdot x \geq \sum_{x \geq k} \Pr[X = x] \cdot x \\ &\geq \sum_{x \geq k} \Pr[X = x] \cdot k = k \Pr[X \geq k] \end{aligned}$$

Deviation Bounds

Markov does not mention variance
But small variance implies concentration, no?

Chebyshev's inequality

For every X and every k

$$\Pr[|X - E[X]| > k] \leq Var(X)/k^2$$

Equivalently, $\Pr[|X - E[X]| \geq k\,\sigma(X)] \leq 1/k^2$

Proof:

$$\begin{aligned} \Pr[|X - E[X]| > k] &= \Pr[(X - E[X])^2 > k^2] \leq \text{ (Markov)} \\ &\leq E[(X - E[X])^2]/k^2 = Var(X)/k^2 \end{aligned}$$

Chebyshev gives (ε, δ)-approximations

$\Pr[|X - E[X]| > k\sigma]$

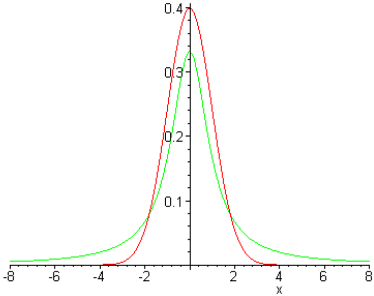
$k = 1$	$k = 2$	$k = 3$	$k = 4$
≤ 1	≤ 0.25	≤ 0.11	≤ 0.07

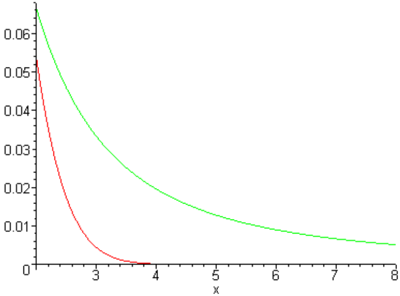
But if X is normally distributed,

$k = 1$	$k = 2$	$k = 3$	$k = 4$
≤ 0.32	≤ 0.05	≤ 0.003	$\leq 3 \cdot 10^{-5}$

Sums of Independent Variables

$\exp(-x^2)$ vs. $1/x^2$:





Sums of Independent Variables

- Suppose $X = \sum_{i=1}^{n} X_i$, $E[X_i] = p$, $Var(X_i) = \sigma^2$, all X_i independent and bounded
- By the Central Limit Theorem, $Z_n = (X - np)/\sqrt{n\sigma^2}$ tends to normal $N(0,1)$ as $n \to \infty$,
- And approximating by the normal gives

$$\Pr[\, Z_n \geq \alpha \,] \approx \exp(-\alpha^2/2)$$

- Chebyshev only gives

$$\Pr[\, Z_n \geq \alpha \,] \leq \frac{1}{\alpha^2}$$

Bernstein Bound

Let:

- $X_1, X_2, \ldots X_n$ be independent random variables,
- $X_i \in [0,1]$, $Var(X_i) = \sigma^2$,
- $X = \frac{1}{n}\sum_{i=1}^{n} X_i$

Bernstein bound

For every $\varepsilon > 0$,

$$\Pr[|X - E[X]| > \varepsilon] < 2\exp\left(-\frac{\varepsilon^2 n}{2\sigma^2 + 2\varepsilon/3}\right)$$

Chernoff-Hoeffding bounds

- $X_1, X_2, \ldots X_n$ be independent random variables,
- $X_i \in [0,1]$, $E[X_i] = p$,
- $X = \sum_{i=1}^{n} X_i$, so $E[X] = pn$

Hoeffding bound (absolute deviation)

$$\Pr[X - pn > \varepsilon n] < \exp(-2\varepsilon^2 n)$$
$$\Pr[X - pn < -\varepsilon n] < \exp(-2\varepsilon^2 n)$$

Chernoff bound (relative deviation)

For $\varepsilon \in [0,1]$,

$$\Pr[X - pn > \varepsilon pn] < \exp(-\varepsilon^2 pn/3)$$
$$\Pr[X - pn < -\varepsilon pn] < \exp(-\varepsilon^2 pn/2)$$

Example: Approximating the Mean

Input: ε, δ, random variable $X \in [0,1]$

Output: (ε,δ)-approximation of $E[X]$

Algorithm $A(\varepsilon,\delta)$

- Draw $n = \dfrac{1}{2\varepsilon^2} \ln \dfrac{2}{\delta}$ copies of X
- Output their average Y

Example: Approximating the Mean

- Let X_i be ith copy of X
- Then $Y = \frac{1}{n}\sum_{i=1}^{n} X_i$, and $E[Y] = E[X]$
- By Hoeffding,

$$\begin{aligned}\Pr[|Y - E[X]| > \varepsilon] &= \Pr[\sum_{i=1}^{n} X_i - E[\sum_{i=1}^{n} X_i] > \varepsilon n] \\ &< 2\exp(-2\varepsilon^2 n) = 2\exp(-\ln(2/\delta)) = \delta\end{aligned}$$

- A different, sequential, algorithm gets (ε, δ) relative approximation using

$$O\left(\frac{1}{\varepsilon^2 E[X]} \ln \frac{1}{\delta}\right)$$

samples of X
[Dagum-Karp-Luby-Ross 95, Lipton-Naughton 95]

Example: Approximating the Median

Input: ε, δ, set $S \subseteq [0,1]$

Output:
an element $s \in S$ whose rank in S is in $(1/2 \pm \varepsilon)|S|$

Algorithm $A(\varepsilon, \delta)$

- Draw $n = \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$ random elements from S
- Output the median of these n elements

Example: Approximating the Median

- Let X_i be 1 if ith sample has rank $\leq (1/2 - \varepsilon)|S|$, 0 otherwise
- $E[X_i] = 1/2 - \varepsilon$
- By Hoeffding,

$$
\begin{aligned}
& \Pr[\geq n/2 \text{ draws give elements with rank} \leq (1/2-\varepsilon)|S|] \\
\leq \quad & \Pr[\sum_{i=1}^{n} X_i \geq n/2] = \Pr[\sum_{i=1}^{n} X_i \geq E[\sum_{i=1}^{n} X_i] + \varepsilon n] \\
\leq \quad & \exp(-2\varepsilon^2 n) = \delta/2
\end{aligned}
$$

- Therefore, with probability $< \delta/2$ we draw $\geq n/2$ elements of rank $\leq (1/2-\varepsilon)|S|$. Implies median of sample $> (1/2-\varepsilon)|S|$
- Similarly the other side

Example use in Data Streams: Sampling rate

Exercise 2.

Understand the algorithm and proof for the median
(You don't have to deliver this exercise, but you have to do it)

Example use in Data Streams: Sampling rate

- Suppose items arrive at so high speed that we have to skip some
- Sample randomly:
  - Choose to process each element with probability α
  - Ignore each element with prob. $1 - \alpha$
- At any time t, if queried for the median, returned the median of the elements chosen so far

Exercise 3.

Given α, δ, determine the probability ε_t such that at time t the output of the algorithm above is an (ε_t, δ)-approximation of the median on the first t elements of the stream