An Algorithmic Excursion in Data Streams

Sudipto Guha
UPenn
What is a Data Stream

Data Items $x_1, x_2, ..., x_n, ...$

Small storage space: sublinear

Example $\sqrt{n}, ..., \log^2 n, ..., k$

Compute $f(x_1, x_2, ..., x_n, ...)$, access $x_i$ in order

Any item not explicitly stored is lost.
Some history


[Munro, Patterson] Selection and sorting in limited storage, 1980.


[Henzinger, Raghavan, Rajagopalan] Computing on Data Streams, 1996.
Points of comparison

- **Online Algs:** No explicit space restriction
  - An Online algorithm with restricted space is a stream alg.

- **Property Testing:** Do not want to inspect whole input

- **Sampling:** [R. Kannan’s talk]

- **Issue:** Can we make more than one pass?
One or Many

- [Munro, Patterson] medians etc. multiple passes

- Cf. S. Rajagopalan’s talk earlier.

- Many passes: The world is a massive data set.

- One pass: Transient data, Network stats.

Can we write to the stream in case of multiple passes?
There is one other ...

- Consider the graphics card in this machine ...
A new hope

- The modern cards are programmable
- They can be used for non-graphics purposes, computing FFT, matrix multiplication etc. etc.
- Computation is speeded up due to (supposed) pixel level parallelism.

- Basically a pipelined SIMD machines. No writes.
- Natural questions in optimization “how many passes do I need to render this class of scenes”.
This talk

- “What are algorithmic ideas in developing data stream algorithms”.
- Will avoid sampling, covered previously.
A Vehicle: Histograms

- How many people in China have more than $100M?

1. Select on country
2. Select on worth

How many Theoretical Computer Scientists?

1. Select on worth
2. Select on country
An example

Lots of uses, point queries, range queries ...

Problem Definition

- Input \( \langle i, f(i) \rangle \langle i+1, f(i+1) \rangle \)
- Find the best piecewise constant approximation minimizing the \( \ell_2^2 \) norm
- \#pieces = B

Many, many other variants possible. We will not get into them.
An observation

- If $B=1$, we want to approximate several $f(i)$ by 1 value :: should use the mean!
- :: over an interval the error will be the variance times the length of interval :: can be computed.

- Now suppose we guessed that the last “bucket” was $[j\ldots n]$ what can we say about the first $B-1$ buckets?
The New Algorithm idea

For each new element $i$
   For $k=1$ to $B$ do
      For a small subset of $j$ (will be $O(\log n)$)
         \[
         E[1\ldots i, k] = \min \left[ E[1\ldots i, k], \ E[1\ldots j, k-1] + E_b(j+1, i) \right]
         \]

$O(Bn)$ space.
$O(n^2B)$ time.
Yes and not exactly

- Can do it for two consecutive k’s
- Picture looks more like...

How to identify the special/interesting split points?

\[
E[1...i, k] = \min \left\{ E[1...i, k], E[1...j,k-1] + Eb(j+1,i) \right\}
\]

When the value of \(E[1..j,k-1]\) changes significantly?
Proof by Picture

- Recall $E[1...i, k] = \text{Min} \left[ E[1...i, k], \ E[1...j,k-1] + Eb(j+1,i) \right]$
- $E[1...j,k-1]$ increases in $j$. Why? more data
- $Eb(j+1,i)$ decreases in $j$. Why? Less..

Approximate in powers of $1 + \frac{\varepsilon}{B}$

For each $k$ : $O(B\varepsilon^{-1} \log n)$

Total space : $O(B^2\varepsilon^{-1} \log n)$

Total time : $O(nB^2\varepsilon^{-1} \log n)$
Do epsilons deceive?

- Not really.
- $B=10$. 

<table>
<thead>
<tr>
<th>number of elements</th>
<th>time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>1.6</td>
</tr>
<tr>
<td>10000</td>
<td>6.7</td>
</tr>
<tr>
<td>20000</td>
<td>36.6</td>
</tr>
<tr>
<td>40000</td>
<td>156.1</td>
</tr>
</tbody>
</table>

For $\epsilon$: 
- $\epsilon=1$: 0.12, 0.3, 0.7, 1.5
- $\epsilon=0.1$: 0.4, 1, 2.7, 7.3
- $\epsilon=0.01$: 2.45, 9.3, 29.6, 86
- $\epsilon=0.001$: 3.7, 19.2, 82.7, 294
Approximate histograms

- [Guha, Koudas, Shim] Data Streams and Histograms, 2001
- Extends to piecewise splines, etc. etc., wherever we can get sum of two such monotone functions ...

- The powers of (1+epsilon/B) approximation has been christened Exponential Histograms and have found use in various other contexts, including sliding window streams by [Datar, Gionis, Indyk, Motwani], 2002.

- But what general tool does this point to re Data Stream Algorithms?
(i) Pruning computation

- Consider the problem of finding the element of rank \( n/2 \) in a stream.
- Build a search tree that stores all the elements.
- Systematically Prune the tree as elements arrive.
- If we stored all elements = exact answer.
- Systematic pruning = approximation guarantee.

- [Greenwald, Khanna] Space efficient online computation of quantile summaries, 2001
Back to Histograms ...

For each new element $i$
   For $K=1$ to $B$ do
      For a small subset of $j$
         $$E[1...i, k] = \text{Min} \left[ E[1...i, k], \ E[1...j,k-1]+E_b(j+1,i) \right]$$

Why are we finding that small subset over and over again?
Approximate Search

For each element i

- Store prefix sums of the elements, squares etc.

To find $E[1..n,k]$

Repeat

- Find $j$ such that $E[1..j,k-1]$ increases geometrically.

$E[1...n, k] = \min \{ E[1...n, k], E[1...j,k-1] + E_b(j+1,n) \}$

Caveat: We do not know the value of $E[1..j,k-1]$, but we know its monotone. We can achieve this approximately and using recursion.
Net result

- $O(n)$ space. But $O(n + B^3 \varepsilon^{-2} \log^2 n)$ time!
(ii) Divide and Conquer


Total Space Required:
A model of clustering

- K medians. Sum of distances.
Thoughts on clustering

- Consider the following move

Is the “move” profitable?
Bicriteria results

- Use \((1+a)k\) medians and guarantee \((1+2/a)\) times the optimum solution. Extends to sum of squares with a weaker guarantee.

- Use the “prices” to control the number of medians. Jain-Vazirani, 1999.

- Also relevant Arya, Garg, Munagala, Pandit local search algorithm on “swaps”.
The times

![Synthetic Stream: CPU Time](chart)

- **BirchKM**
- **StreamKM**
- **BirchLS**
- **StreamLS**
And the quality

![Synthetic Stream: SSQ](image)
**Histograms, similar story...**

- In this case the partitioning does not hold

---

New Algorithm

Old algorithm (not quite, but in spirit)
To finish up ...

- Space=$M$, time $O(n+(n/M)B^3 \varepsilon^{-2} \log^2 n)$
- If $M=O(B^3 \varepsilon^{-2} \log^2 n)$ then amortized $O(1)$ per element.
- To appear in journal version of [GKS]

<table>
<thead>
<tr>
<th></th>
<th>Eps=1</th>
<th>Eps=0.5</th>
<th>Eps=0.1</th>
<th>Eps=0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>N=10^6</td>
<td>1.7</td>
<td>1.8</td>
<td>23</td>
<td>81.4</td>
</tr>
<tr>
<td>N=10^7</td>
<td>14.6</td>
<td>17.5</td>
<td>83</td>
<td>294</td>
</tr>
<tr>
<td>N=10^8</td>
<td>162</td>
<td>171</td>
<td>652</td>
<td>1841</td>
</tr>
</tbody>
</table>

800 MB, 15 mins to generate

$M=100000$, time in seconds
(iii) Composition

- Implicit “writes” to a stream.
- Not true always …
A twist to the model

- What happens if …\(i, f(i), i+1, f(i+1)\)…… order is not preserved? For example we get
  \[5, f(5), 5000, f(5000), 10, f(10)\] (an unsorted database)

- A more general model: stream of transactions
  … (1 beer)…(2 diapers)…(one more beer)…(17 more beers)

- [FM] [AMS] work over this model.
(iv) Embeddings

An old story:
"Why are you searching here?"
"The light is here."

Transform the problem to a different problem which can be solved more easily.
Embeddings contd.

- Mapped to diff domain.
- A possibly different problem is solved in the new domain.
- Map back the solution.

Cf. Kernel methods...

\[ \text{Data} \xrightarrow{\Phi} \Phi(\text{Data}) \xrightarrow{g} g(\Phi(\text{Data})) \xrightarrow{f} f(\text{Data}) \]
Johnson Lindenstrauss Lemma

- Given a matrix $A$ whose elements are iid Gaussian, and any vector $x$, with high prob.

\[
\|x\|_2 \leq \|Ax\|_2 \leq (1 + \varepsilon)\|x\|_2
\]

if $x \in \mathbb{R}^n$ then $A \in \mathbb{R}^{O(\log n) \times n}$

$\Rightarrow Ax \in \mathbb{R}^{O(\log n)}$.

Dimensionality reduction, nearest neighbor.
Linear Embeddings

\[ y_i = a_{i1}x_1 + \cdots a_{ij}x_j + \cdots a_{in}x_n = \langle a_i, x \rangle \]

- To update \( x_j \) by \( +a \)
  \[ y'_i = a_{i1}x_1 + \cdots a_{ij}(x_j + a) + \cdots a_{in}x_n \]

- But this is easy: \( y'_i = y_i + a_{ij}a \) !!!
- So to update \( Ax \), we do not need to store \( x \).
- How to store \( O(n \log n) \) elements of the matrix ??
Generate on the fly

- Imagine the matrix is generated by a pseudorandom generator and we store the seed. Every time we want $a_{ij}$ we generate it!
- [Indyk 2000]

- Achieved differently by different papers. Limited independence hash functions is one way. Codes ...
- [Feigenbaum, Kannan, Strauss, Vishwanathan 1999]
An example: SVD

- Given \( m \) streams \( X \) (rows are streams) compute the best possible correlation

- Basically \( \max ||y^T X|| \)

- Hmm. \( ||y^T X|| \leq ||y^T X A^T|| \leq (1 + \epsilon)||y^T X|| \)

- Maximize \( ||y^T X A^T|| \) ?? Need to store \( X A^T \)
- This requires \( O(m \log n) \) space, weaker than results mentioned in sampling talk, but in a transactions model...
- Precision is an issue.
Back to histograms

- Suppose we were given the boundaries ...
- Denote by $\tilde{v}$ the heights of these intervals
- Now $\|x - h\|_2 \approx_\varepsilon \|Ax - Ah\|_2$
- But $Ah$ is a linear function of $\tilde{v}$
- Minimization possible.
How to get the boundaries?

1. Focus on dyadic intervals
2. Focus on intervals of same length
3. Suppose some $B'$ of them contain $1/\log n$ fraction of the "energy" of the signal.
4. A Top $B'$ query: for this set of intervals.
5. Take them out, recurse.

We know $Ax$, therefore $Ax-Ah$ gives us $A(x-h)$
Why does it work?

- One of two will happen:
  1. Either we will approximate signal well
  2. The error does not decrease => nobody can approximate better.

- Either case we approximate the error.
Top k?

- Finds all elements present with frequency 1/k.
- Say element i has freq 1/2
- We can get the bits of i.
- In case of general k choose a k-wise indep. Permutation. WHP only one element element with large frequency is in [0..n/k], and other elements cumulatively have low freq. Perform the above.
- Group testing. But how does the k-wise permutation, etc. work with sketches ... it does.
Putting things together

- Maintain sketch $Ax$ of the stream $x$.
- Find the boundaries. (Recursively subtract the dyadic intervals with highest energy - use [recursively maintained] sketch to estimate the energy)
- Use the boundaries and the sketch to construct histogram.

- Poly($B, \log n$) space, poly($B, \log n$) time per element.
Recap

- Approximation algorithms for data streams feasible. Moreover there is some structure in designing them.

- Faced with small space we:
  1. Order the following from Amazon.
  2. Sample
  3. Prune computation
  4. Embed
  5. Divide and Conquer
  6. Mix and Match - compose