

An Algorithmic Excursion in Data Streams

Sudipto Guha
UPenn



What is a Data Stream

Data Items $x_1, x_2, \dots, x_n, \dots$

Small storage space : sublinear

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

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

Any item not explicitly stored is lost.

Some history

[Morris] Counting large number of events in small registers, 1978.

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

[Flajolet, Martin] Probabilistic counting, 1983.

[Alon, Matias, Szegedy] The space complexity of approximating the frequency moments, 1996.

[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

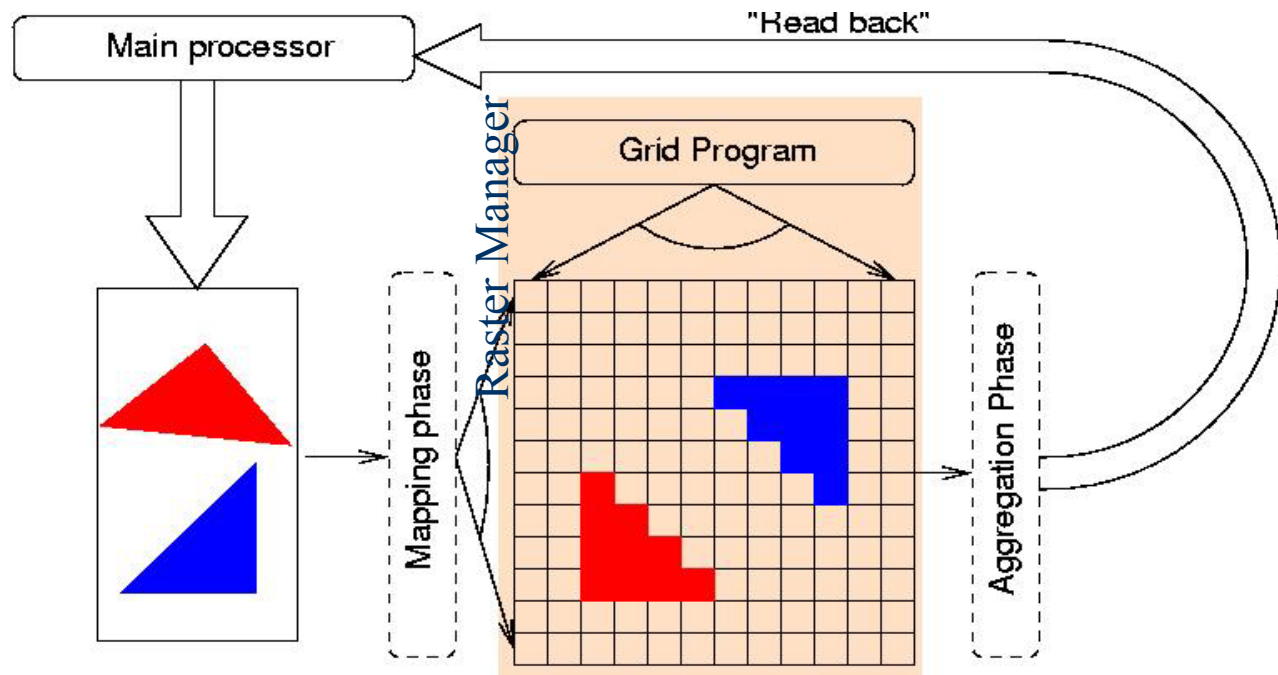
- [Munro, Patterson] medians etc. multiple passes
- Cf. S. Rajagopalan's talk earlier.
- Many passes: The world is a massive data set.
- One pass: Transient data, Network stats.

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.

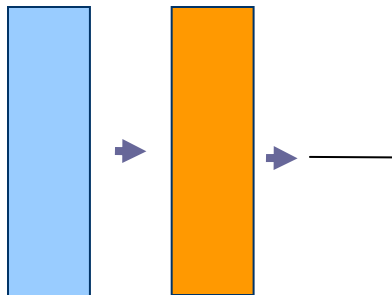
- Basically a pipelined SIMD machines. No writes.
- Natural questions in optimization "how many passes do I need to render this class of scenes".

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 ?



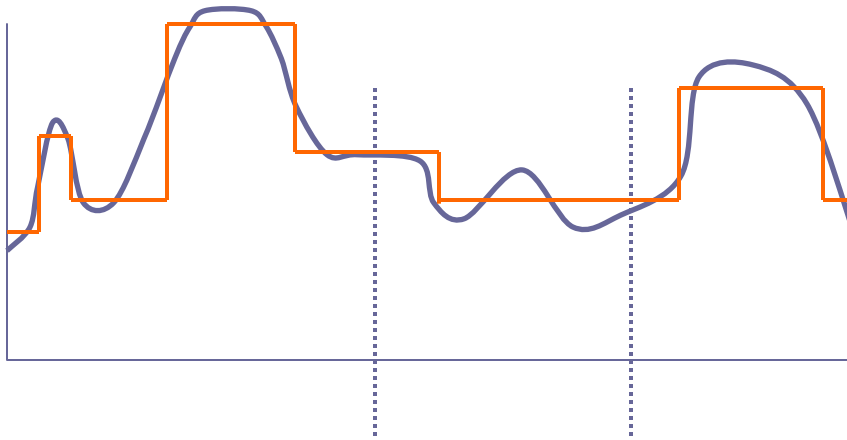
1. Select on country
2. Select on worth

How many
Theoretical
Computer
Scientists ?



1. Select on worth
2. Select on country

An example

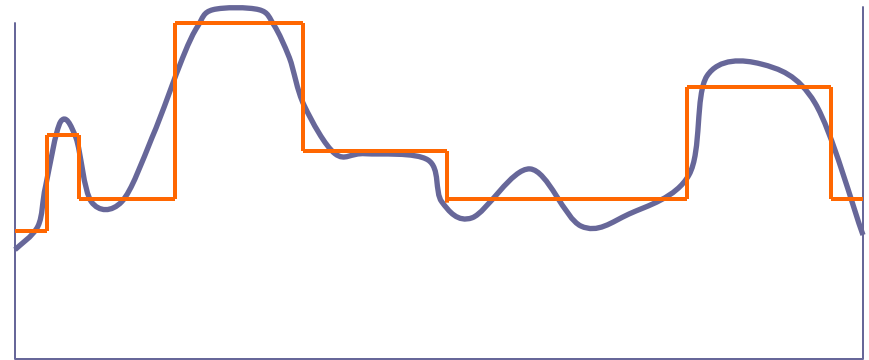


Lots of uses, point queries, range queries ...

AQUA : Gibbons, Matias, and Poosala, 1997.

Problem Definition

- Input $\langle i, f(i) \rangle \langle i+1, f(i+1) \rangle$
- Find the best piecewise constant approximation minimizing the ℓ_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 \therefore should use the mean!
- \therefore over an interval the error will be the variance times the length of interval \therefore can be computed.
- Now suppose we guessed that the last "bucket" was $[j\dots n]$ what can we say about the first $B-1$ buckets ?

The New Algorithm idea

For each new element i

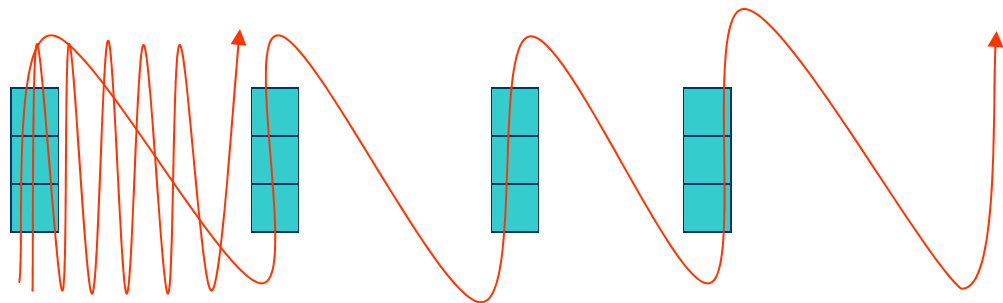
For $K=1$ to B do

For a small subset of j (will be $O(\log n)$)

$$E[1..i, k] = \text{Min} [E[1..i, k], E[1..j, k-1] + E_b(j+1, i)]$$

$O(Bn)$ space.

$O(n^2B)$ time.



Proof by Picture

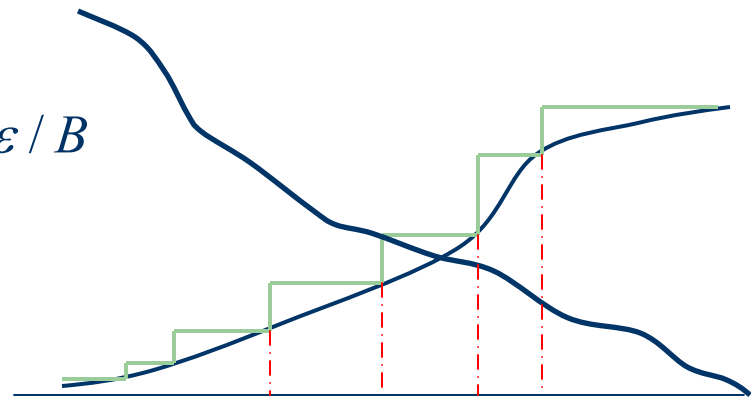
- Recall $E[1\dots i, k] = \text{Min} [E[1\dots i, k], E[1\dots j, k-1] + E_b(j+1, i)]$
- $E[1\dots j, k-1]$ increases in j . Why? more data
- $E_b(j+1, i)$ decreases in j . Why? Less..

Approximate in powers of $1 + \varepsilon / B$

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

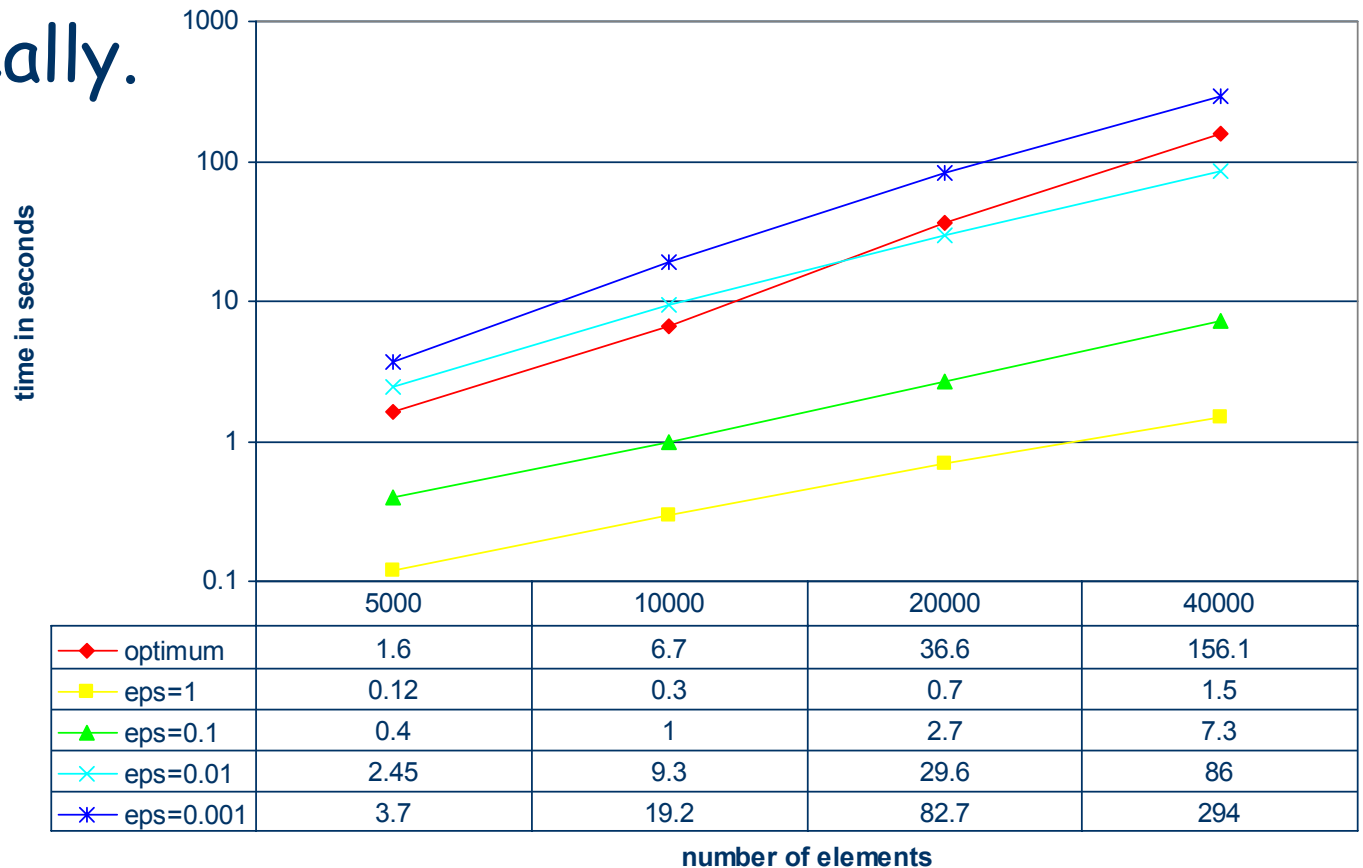
Total space : $O(B^2\varepsilon^{-1} \log n)$

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



Do epsilons deceive ?

- Not really.
- $B=10$.



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

$$E[1\dots i, k] = \text{Min} [E[1\dots i, k], E[1\dots j, k-1] + E_b(j+1, i)]$$

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

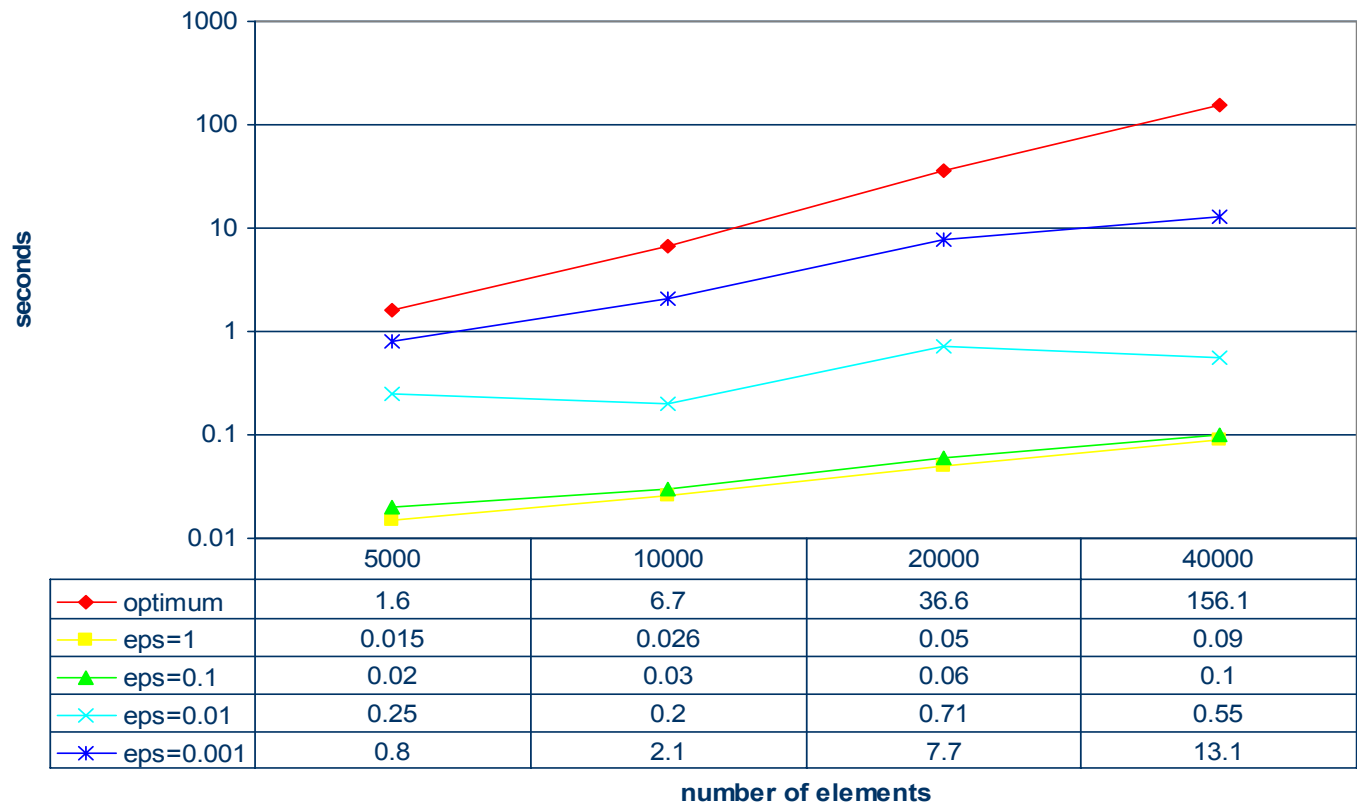
Find j such that $E[1..j,k-1]$ increases geometrically.

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

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.

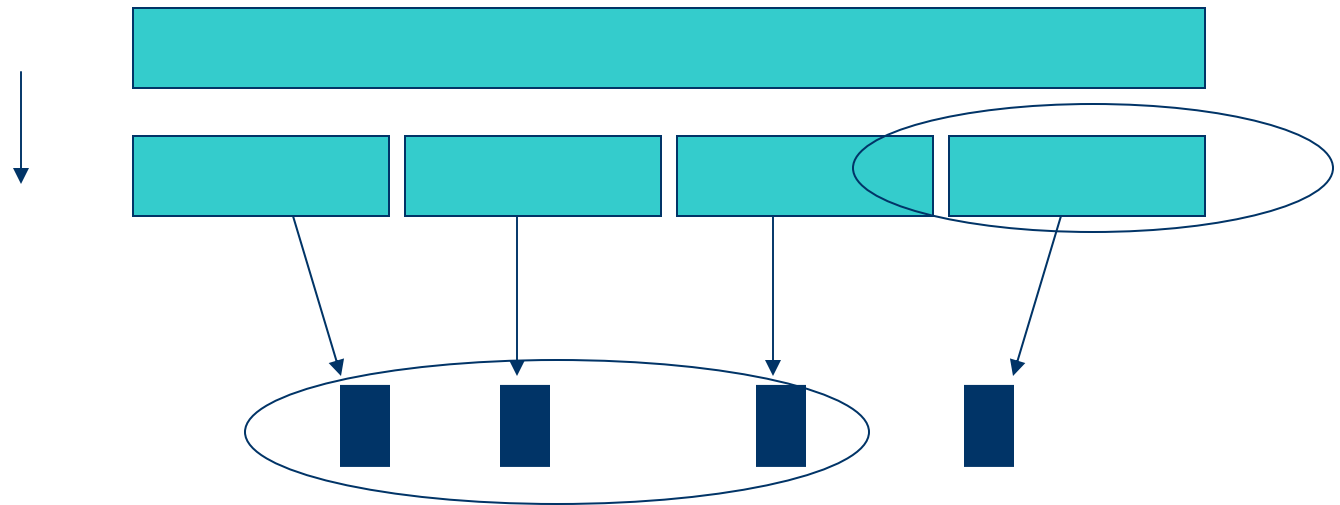
Net result

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



(ii) Divide and Conquer

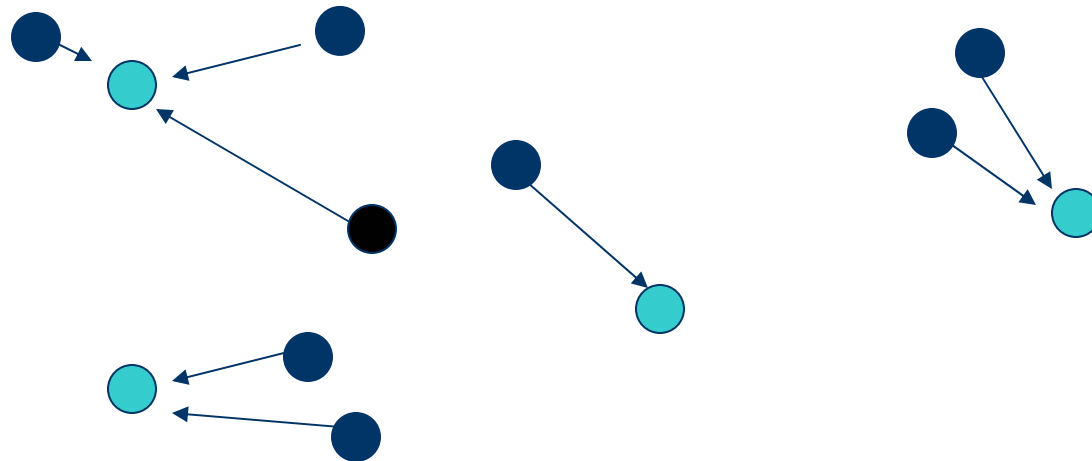
[Guha, Mishra, Motwani, O'callaghan] Clustering Data Streams, 2000.
Improvements in [Charikar, Panigrahy, O'callaghan] STOC 2003.



Total Space Required: 

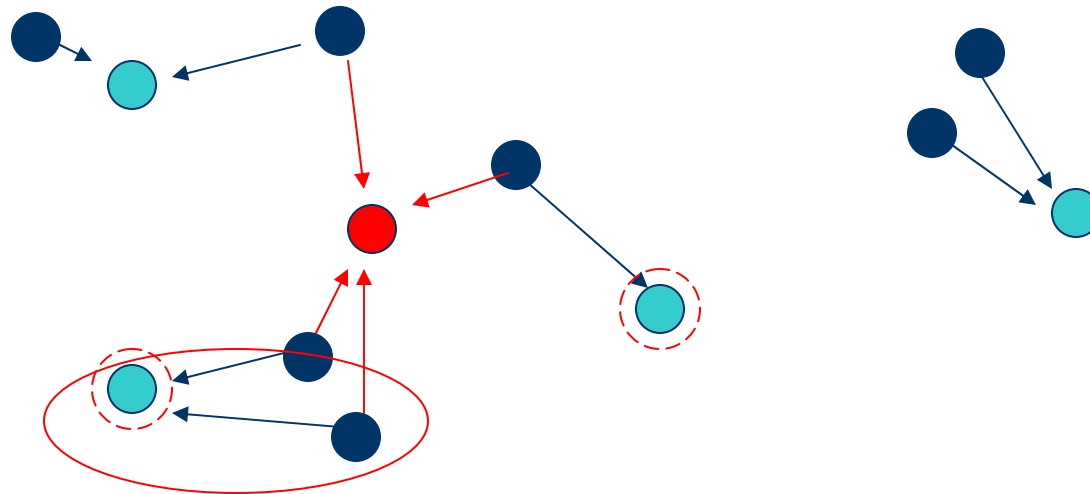
A model of clustering

- K medians. Sum of distances.



Thoughts on clustering

- Consider the following move

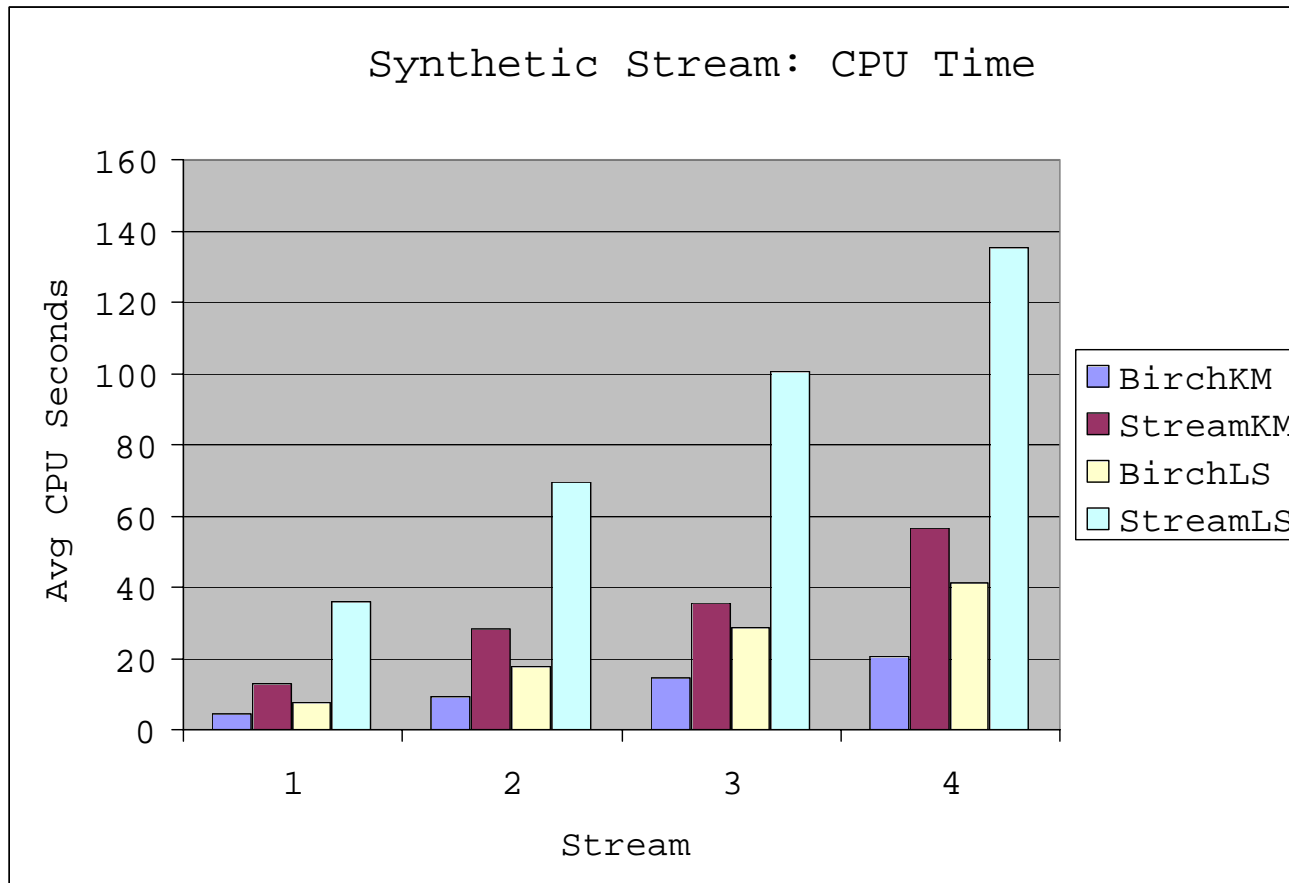


Is the "move" profitable?

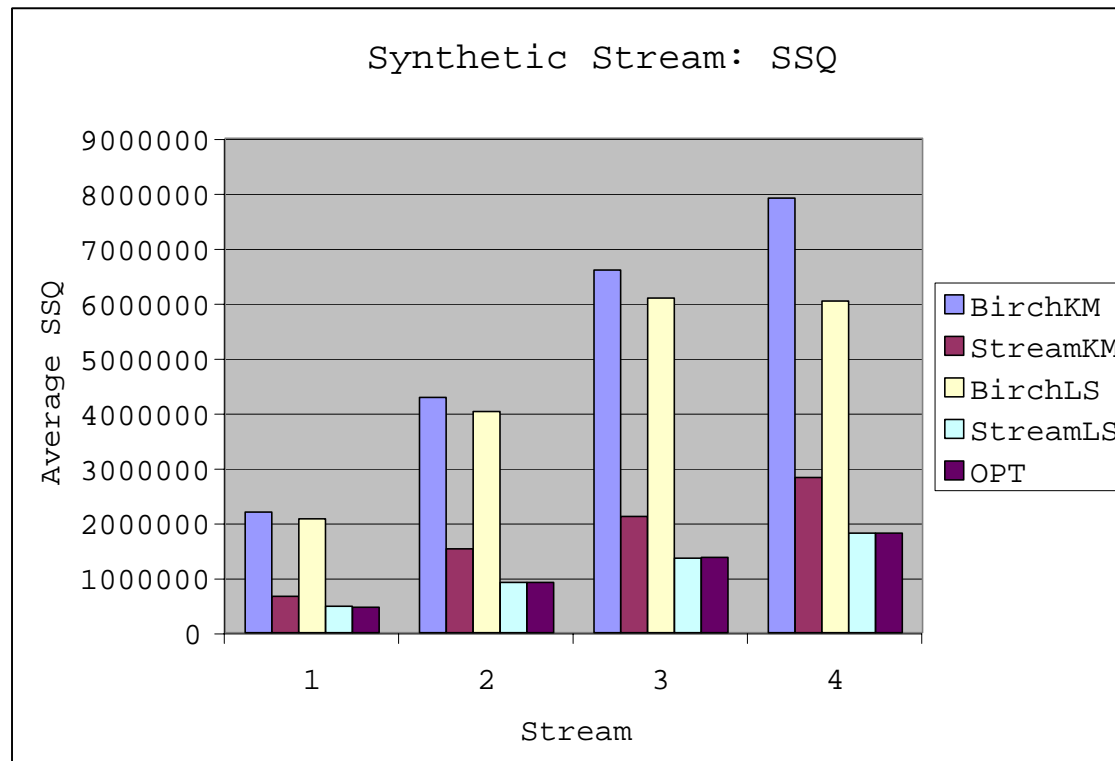
Bicriteria results

- 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

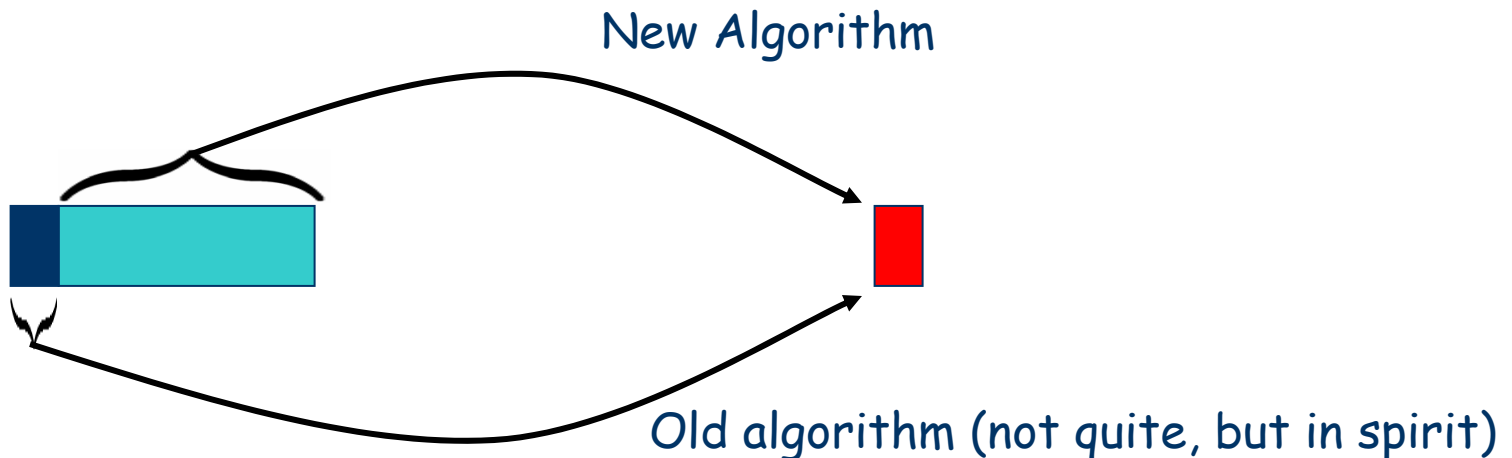


And the quality



Histograms, similar story...

- In this case the partitioning does not hold



To finish up ...

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

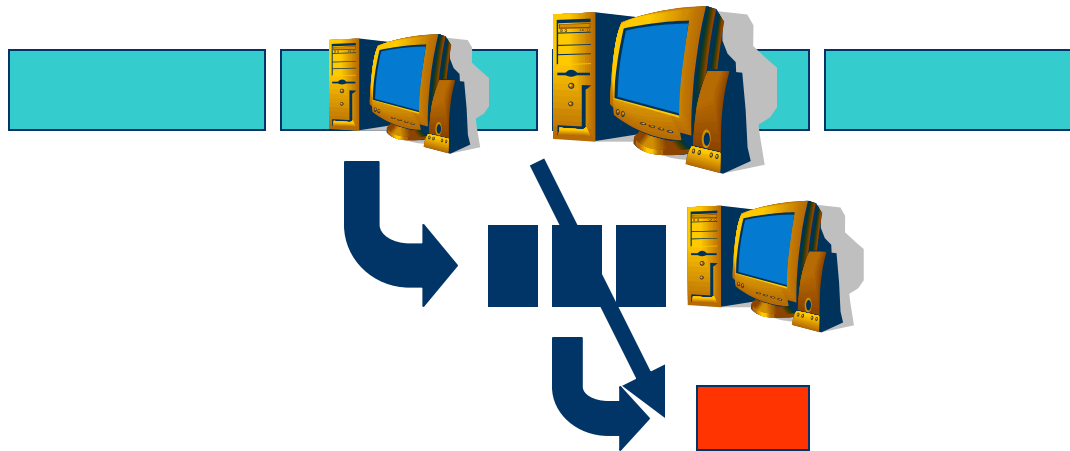
	Eps=1	Eps=0.5	Eps=0.1	Eps=0.05
$N=10^6$	1.7	1.8	23	81.4
$N=10^7$	14.6	17.5	83	294
$N=10^8$	162	171	652	1841

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 $\dots \langle i, f(i) \rangle, \langle i+1, f(i+1) \rangle \dots$ order is not preserved? For example we get $\langle 5, f(5) \rangle \dots \langle 5000, f(5000) \rangle \dots \langle 10, f(10) \rangle$ (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?"

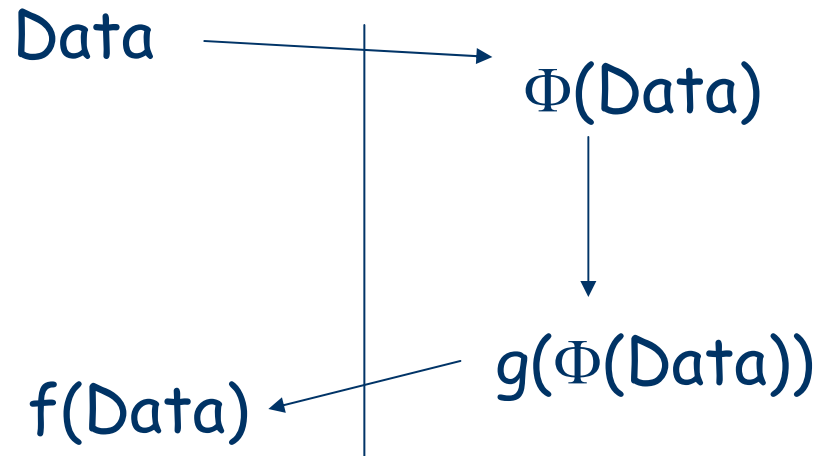
"The light is here."

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...



Johnson Lindenstrauss Lemma

- Extensions of Lipschitz mapping to Hilbert Spaces, 1984.
- 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 $x \in R^n$ then $A \in R^{O(\log n) \times n}$
 $\Rightarrow Ax \in R^{O(\log n)}$.

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

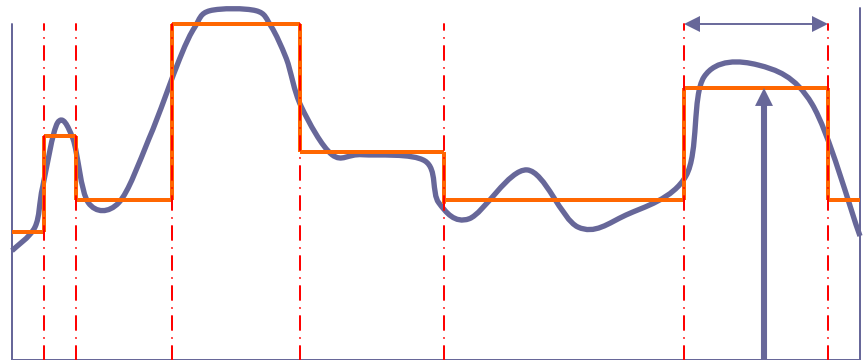
- Imagine the matrix is generated by a pseudorandom generator and we store the seed. Every time we want a_{ij} we generate it!
- [Indyk 2000]
- Achieved differently by different papers. Limited independence hash functions is one way. Codes ...
- [Feigenbaum, Kannan, Strauss, Vishwanathan 1999]

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 \vec{v} the heights of these intervals
- Now $\|\mathbf{x} - \mathbf{h}\|_2 \approx_\varepsilon \|\mathbf{Ax} - \mathbf{Ah}\|_2$
- But \mathbf{Ah} is a linear function of \vec{v}
- Minimization possible.

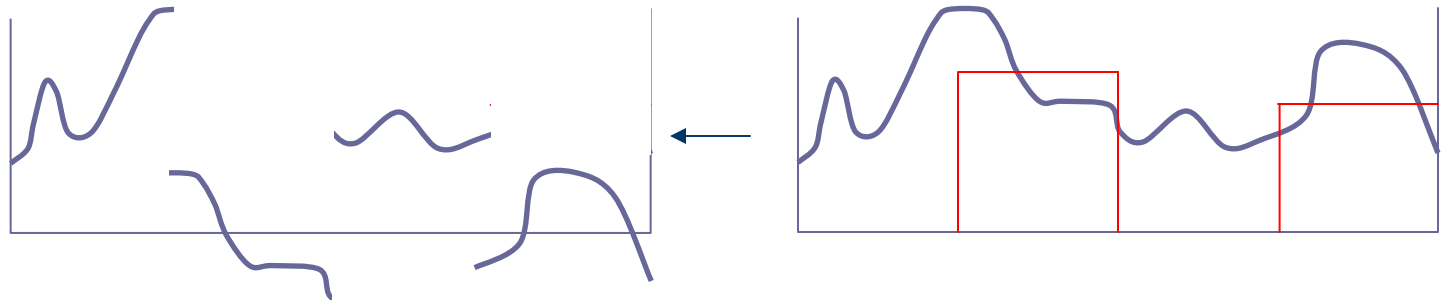


How to get the boundaries ?

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 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.

- $\text{Poly}(B, \log n)$ space, $\text{poly}(B, \log n)$ time per element.
- [Gilbert, Guha, Indyk, Kotidis, Muthukrishnan, Strauss] 2002.

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

