Fast Approximate Wavelet Tracking on Streams

Graham Cormode
cormode@bell-labs.com

Minos Garofalakis
minos.garofalakis@intel.com

Dimitris Sacharidis
dsachar@dblab.ntua.gr

outline

- introduction
  - motivation
  - problem formulation
- background
  - wavelet synopses
  - the AMS sketch
- the GCS algorithm
  - our approach
  - the Group Count Sketch
  - finding L_2 heavy items
  - sketching the wavelet domain
- experimental results
- conclusions

motivation

- numerous emerging data management applications require to continuously generate, process and analyze massive amounts of data
  - e.g. continuous event monitoring applications: network-event tracking in ISPs, transaction-log monitoring in large web-server farms
- the **data streaming** paradigm
  - large volumes (~Terabytes/day) of monitoring data arriving at high rates that need to be processed on-line
- analysis in data streaming scenarios rely on building and maintaining approximate synopses in real time and in one pass over streaming data
  - require **small space** to summarize key features of streaming data
  - provide **approximate** query answers with **quality guarantees**

problem formulation

- our focus: maintain a wavelet synopsis over data streams
- algorithmic requirements:
  - **small memory footprint** (sublinear in data size)
  - **fast per stream-item process time** (sublinear in required memory)
  - **fast query time** (sublinear in data size)
  - **quality guarantees** on query answers

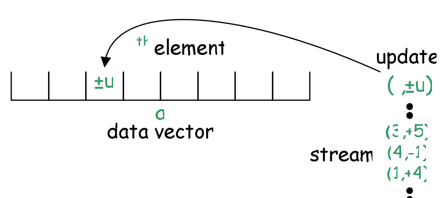
stream processing model

assume data vector a of size N

stream items are of the form (i,±u)

denoting a net change of ±u in the a[i] entry

a[i] := a[i] ± u



interpretation

u insertions/deletions of the i^{th} entry (we also allow entries to take **negative** values)

important

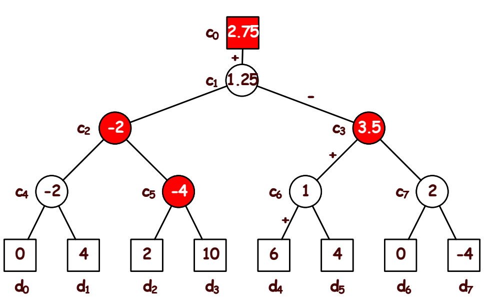
items are only seen **once** in the fixed order of arrival and **do not** come ordered in i

outline

- introduction
  - motivation
  - problem formulation
- background
  - wavelet synopses
  - the AMS sketch
- the GCS algorithm
  - our approach
  - the Group Count Sketch
  - finding L_2 heavy items
  - sketching the wavelet domain
- experimental results
- conclusions

wavelet synopses

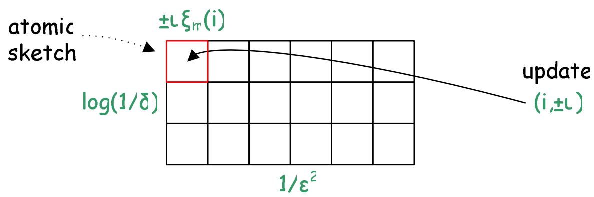
- the (Haar) **wavelet decomposition** hierarchically decomposes a data vector
  - for every pair of consequent values, compute the **average** and the semi-difference (a.k.a. **detail**) values (coefficients)
  - **iteratively** repeat on the **lower-resolution** data consisting of only the averages
  - final decomposition is the overall average plus all details



- to obtain the optimal, in sum-squared-error sense, wavelet synopsis only keep the highest in absolute normalized value coefficients
  - implicitly set other coefficients to zero
- easily extendable to multiple dimensions

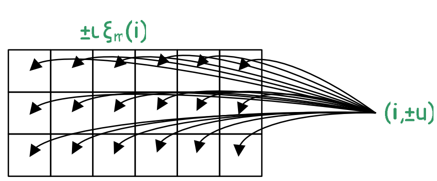
the AMS sketch (1/2)

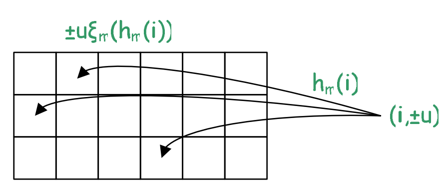
- the **AMS sketch** is a powerful data stream synopsis structure serving as the building block in a variety of applications:
  - e.g.: estimating (multi-way) join size, constructing histograms and wavelet synopses, finding frequent items and quantiles
- it consists of $O(1/\varepsilon^2) \times O(\log(1/\delta))$ atomic sketches
- an atomic AMS sketch X of a is a randomized linear projection
  - $X = \langle a,\xi\rangle = \Sigma_i\, a[i]\xi(i)$, where ξ denotes a random vector of four-wise independent random variables $\{\pm 1\}$
  - the random variable can be generated in just $O(\log N)$ bits space for seeding, using standard pseudo-random hash functions
- X is updated as stream updates $(i, \pm u)$ arrive: $X := X \pm u\xi(i)$



the AMS sketch (2/2)

- the AMS sketch estimates the L_2 norm (energy) of a
  - let Z be the $O(\log(1/\delta))$-wise median of $O(1/\varepsilon^2)$-wise means of the **square** of independent atomic AMS sketches
  - then Z estimates $||a||^2$ within $\pm\varepsilon||a||^2$ (w.h.p. $\geq 1-\delta$)
  - it can also estimate inner products
- an improvement: fast AMS sketch
  - introducing a **level of hashing** reduces update time by $O(1/\varepsilon^2)$ while providing the same guarantees and requiring same space





outline

- introduction
  - motivation
  - problem formulation
- background
  - wavelet synopses
  - the AMS sketch
- the GCS algorithm
  - our approach
  - the Group Count Sketch
  - finding L_2 heavy items
  - sketching the wavelet domain
- experimental results
- conclusions

our approach (1/3)

two shortcomings of existing approach [GKMS] (using AMS sketches):

1. updating the sketch requires O(|sketch|) updates per streaming item
2. querying for the largest coefficients requires superlinear $\Omega(N\log N)$ time (even when using range-summable random variables)
   - blows up in the multi-dimensional case

can we fix it? use the fast-AMS sketch to speed up update time (not enough)

we introduce the GCS algorithm that satisfies **all** algorithmic requirements
- makes summarizing large multi-dimensional streams feasible

streaming requirements	GKMS	fast-GKMS	GCS
small **space**	✓	✓	✓
fast **update time**	✗	✓	✓
fast **query time**	✗	✗	✓

our approach (2/3)

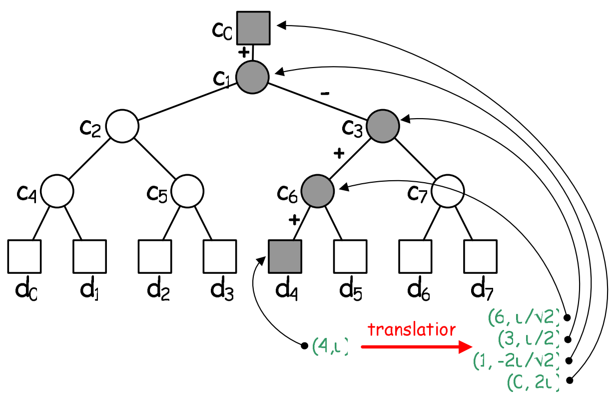
the GCS algorithm relies on two ideas:

(1) sketch the wavelet domain

(2) quickly identify large coefficients

(1) is easy to accomplish: translate updates in the original domain to updates in the wavelet domain

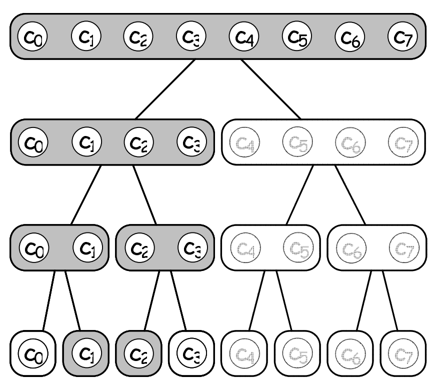
- just polylog more updates are required, even for multi-d



our approach (3/3)

for (2) we would like to perform a binary-search-like procedure

- enforce a **hierarchical grouping** on coefficients
- **prune** groups of coefficients that are not **L_2-heavy**, as they may not contain L_2-heavy coefficients
- only the remaining groups need to be examined more closely
- iteratively keep pruning until you reach singleton groups



but, how do we estimate the L_2 (energy) for groups of coefficients?

- this is a difficult task, requiring a novel technical result
- **more difficult** than finding frequent items!

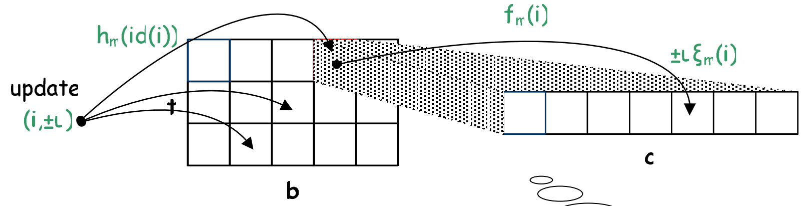
enter group count sketch

group count sketch (1/2)

goal: estimate the L_2 of all k groups forming a partition on the domain of a

the group count sketch (GCS) consists of b buckets each having c sub-buckets, repeated t times

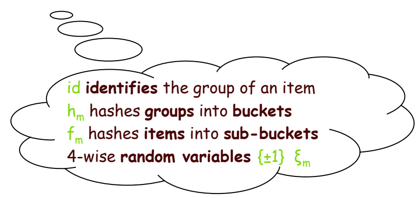
this gives a total of t·b·c counters s[1][1][1] through s[t][b][c]



update the sketch per stream element (i, ±u)

repeat t times:

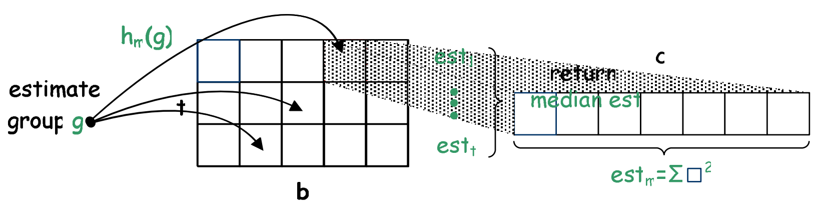
get item's group -> hash into a bucket -> hash into a sub-bucket -> update counter by {±1}·u



group count sketch (2/2)

estimate L_2 of group g

return the median of m instances of $\Sigma_j\,(s[m][h_m(g)][j])^2$ for all j in [c]



estimates are (w.h.p. 1-δ) within additive error $\varepsilon\cdot||a||^2$

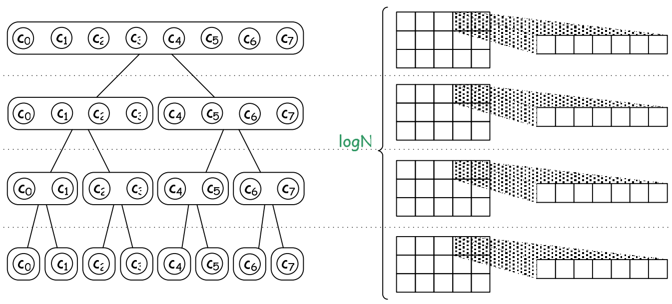
analysis results: $t=O(\log(1/\delta))$ $b=O(1/\varepsilon)$ $c=O(1/\varepsilon^2)$

- space $O(1/\varepsilon^3 \log(1/\delta))$ counters
- update cost $O(\log(1/\delta))$
- query cost $O(1/\varepsilon^2 \log(1/\delta))$

finding L_2-heavy items

keep one GCS per level of hierarchy

space and update time complexities increase (roughly) by a factor of logN



query: find all items with L_2 greater than $\varphi||a||^2$

query time increases by $1/\varphi \cdot \log N$ ($1/\varphi$ L_2-heavy items per level)

w.h.p. we get **all items** with L_2 **greater** than $(\varphi+\varepsilon)||a||^2$

w.h.p. we get **no items** with L_2 **less** than $(\varphi-\varepsilon)||a||^2$

sketching the wavelet domain

the GCS algorithm:

- translate updates into the wavelet domain
- maintain $\log_r N$ group count sketches
- find L_2 heavy coefficients with energy above $\varphi||a||^2$

note: changing the degree (r) of the search tree allows for query-update time trade-off

but, what should the threshold φ be?

assuming the data satisfies the small-B property:

there is a B-term synopsis with energy at least $\eta||a||^2$

setting $\varphi = \varepsilon\eta/B$ we obtain a synopsis (with no more than B coeffs) with energy at least $(1-\varepsilon)\eta||a||^2$

outline

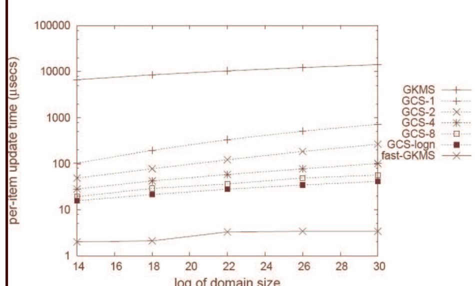
- introduction
  - motivation
  - problem formulation
- background
  - wavelet synopses
  - the AMS sketch
- the GCS algorithm
  - our approach
  - the Group Count Sketch
  - finding L_2 heavy items
  - sketching the wavelet domain
- experimental results
- conclusions

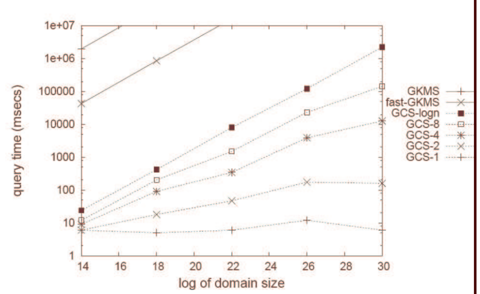
experiments

update and **query** time vs domain size

all methods are given same space

GCS-r is GCS with search tree of degree 2^r

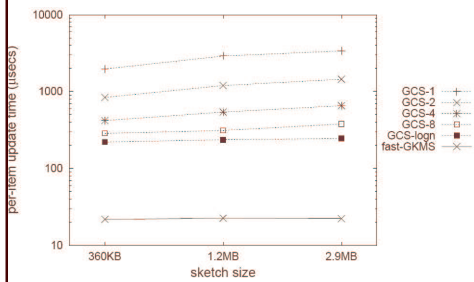


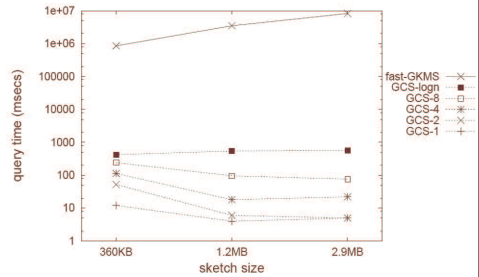


experiments

update and **query** time vs sketch size

GCS-r is GCS with search tree of degree 2^r

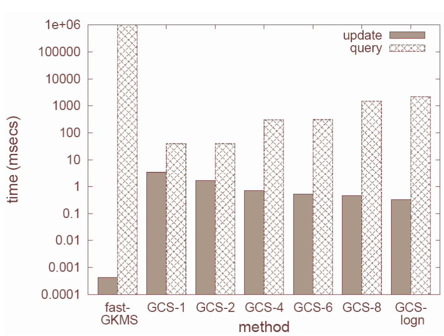




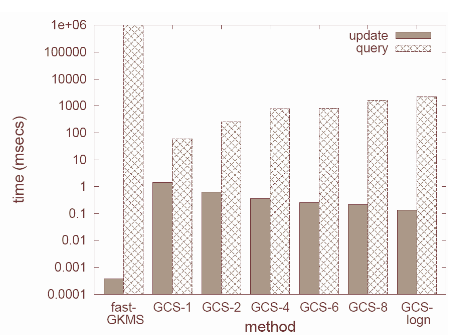
experiments

two-dimensional **update** and **query** time

for both wavelet decomposition forms



standard form



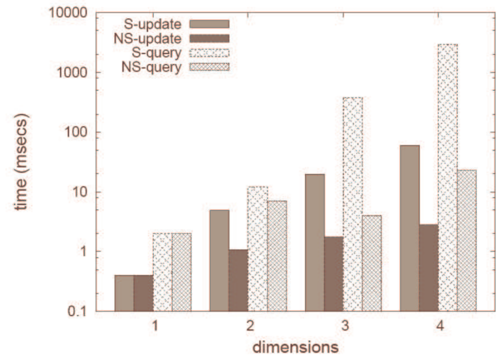
non-standard form

experiments

multi-dimensional **update** and **query** time

for both wavelet decomposition forms

S: standard NS: non-standard



conclusions

- the GCS algorithm allows for efficient tracking of wavelet synopses over multi-dimensional data streams
- the Group Count Sketch satisfies all streaming requirements:
  - small polylog **space**
  - fast polylog **update time**
  - fast polylog **query time**
  - approximate answers with **quality guarantees**
- future research directions:
  - other error metrics
  - histograms

thank you!

http://www.dblab.ntua.gr/