Continuously Maintaining Order Statistics Over Data Streams

Lecture Notes

COM9314

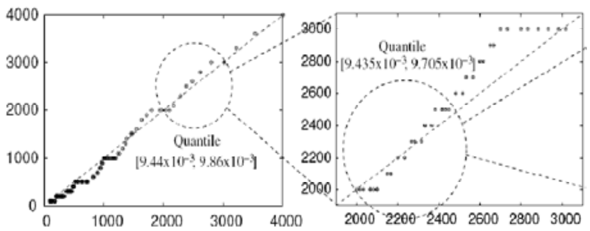
Outline

- Introduction
- Uniform Error techniques
- Relative Error techniques
- Duplicate-insensitive techniques
- Miscellaneous
- Future Studies

Applications

Φ-quantile:
Given Φ∈(0, 1], find the element with rank ⌈ΦN⌉.

Q-Q Plot



Applications

- Equal Width Histograms:

  (x1, 1), (x2, 2), (x3, 3), (x4, **4**), (x5,5), (x6, 6), (x7, 7), (x8, 8), (x9, 9), (x10, 10), (x11, 10), (x12, 10), (x13, 11), (x14, 11), (x15, 11), (x16, 12)

  Support approximate range aggregates.
- In stock market, road traffic, network, given a value, find its rank (or quantile).
- Portfolio risk management counting
- Counting Inversions in on-line Rank Aggregation
- etc.

Rank/Order-based Queries

Given : a set of N data elements (x, v) where v=f(x) and the elements are ranked against a monotonic order of v.

- Rank Query 1 (RQ1):
  Given r, find an element value with the rank r.
- Φ-quantile (a popular form of RQ1)
  Given Φ∈(0, 1], find the element with rank ⌈ΦN⌉.
- Rank Query 2 (RQ2):
  Given v, find how many elements with values less than v.

Note: RQ1 is equivalent to RQ2.

Example

Data Stream:

12, 10, 11, 10, 1, 10, 11, 9, 6, 7, 8, 11, 4, 5, 2, 3

Sorted Sequence:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 11, 11, 11, 12

r=4 (0.25-quantile)

r=8 (0.5 -quantile)

r=12 (0.75 -quantile)

Some Background

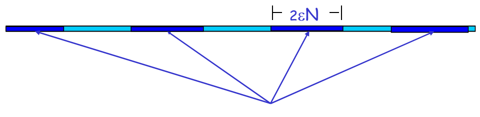
- $O(N^{1/p})$ memory space is required in exact computation in p scans of data [TCS80]
- In data streams
  - *One pass scan*
  - summary with small memory space
- In stream processing, approximation is a good alternative to achieve scalability.

Uniform Error Techniques

- Uniform Error: ϵ-approximate

Given r, return any element e with rank r' within $[r - \varepsilon N, r + \varepsilon N]$ $(0 < \varepsilon < 1)$.

Space Lower bound : $O(1/\varepsilon)$



Uniform Error Technique

- GK Algorithm
- Randomize Algorithm
- Count-Min Algorithm
- Sliding window techniques

GK Algorithm [sigmod01, PSU]

Deterministic Algorithm:
Keep $(v_i, r_{min}(v_i), r_{max}(v_i))$ for each observation i.

Theorem 1: If $(r_{max}(v_{i+1}) - r_{min}(v_i) - 1) < 2\varepsilon N$, then ε-approximate summary.

Tuple: $\{v_i, g_i, \Delta_i\}$; $g_i = r_{min}(v_i) - r_{min}(v_{i-1})$,
$\Delta_i = r_{max}(v_i) - r_{min}(v_i)$

$r_{min}(v_i)$: minimum possible rank of v_i
$r_{max}(v_i)$: maximum possible rank of v_i

GK Algorithm [sigmod01, PSU]

***Goal:* always maintain ε-approximate summary**
$(r_{max}(v_{i+1}) - r_{min}(v_i) - 1) = (g_i + \Delta_i - 1) < 2\varepsilon N$

Insert new observations into summary:
-Insert tuple before the *i*th tuple. $g_{new} = 1$; $\Delta_{new} = g_i + \Delta_i - 1$;

Delete all "superfluous" entries $g_i := g_i + g_{i-1}$ -1

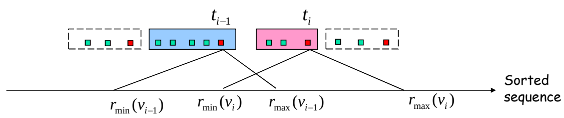
General strategy:

-Delete tuples with small capacity and preserve tuples with large capacity.

- Do batch compression.

GK Algorithm [sigmod01, PSU]

Synopsis structure S : sequence of tuples $t_1, t_2, \ldots, t_s$ where $t_i = (r_{min}(v_i), r_{max}(v_i), v_i)$



Space bound: $O(\frac{1}{\varepsilon}\log \varepsilon N)$ to achieve ε-approximation.

Given r , there's at least one element such that
$r_{max}(v_i) - \varepsilon n \le r \le r_{min}(v_i) + \varepsilon n$
Query alg: first hit.

Randomize Algorithm
[Sigmod99, IBM]

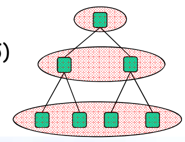
Sampling

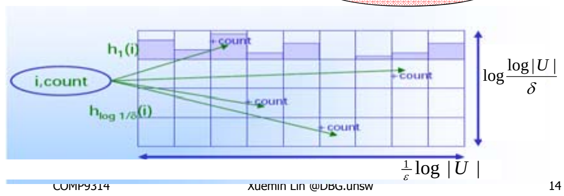
- Exponential reduction of sampling rate regarding an increment of N
- ϵ-approximate with confidence 1-δ
- Feed GK-like (compress) algorithm the samples
- Space bound: $O(\frac{1}{\varepsilon}\log 1/\varepsilon \log 1/(\varepsilon\delta))$

Count-min sketch
[LATIN04, Rutgers Uni]

- Stream with Updates
- ϵ-approximate (confidence 1-δ)
- Space $O(\frac{1}{\epsilon}\log^2|U|\log\frac{\log|U|}{\delta})$
- Basic idea:

Dyadic range





Sliding window technique

Sliding window : the most recent N elements in data streams.

Problem:

Input: data stream D & a sliding window (N)

Output: an ε-approximate quantile summary for the sliding window (N).

Example

Data Stream:

12, 10, 11, 10, 1, 10, 7, 9, 6, 11, 8, 11, 4, 5, 2

Median (in ordered set) — A sliding window (N=9) — Current item

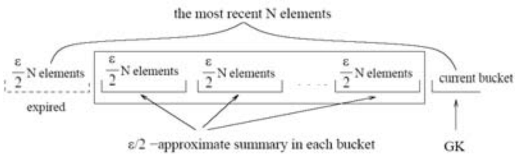
After "3" arrived:

12, 10, 11, 10, 1, 10, 7, 9, 6, 11, 8, 11, 4, 5, 2, 3

Expired elements — Median (in ordered set) — Current item

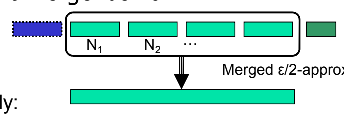
Algorithm [icde04, UNSW]

- Algorithm outline:
  - Partition sliding window equally into $\lceil 2/\varepsilon \rceil$ buckets
  - Maintain an $\frac{\varepsilon}{4}$-approx. sketch in the most recent bucket by GK-algorithm
  - Compress the sketch when the most recent bucket is full.
  - Expire the oldest bucket once a new bucket starts.
  - Space required : $O(\frac{1}{\varepsilon}\log \varepsilon^2 N + \frac{1}{\varepsilon^2})$



Global ε-approximate sketch

- *Step 1:* Merge the compressed sketches in a sort-merge fashion



Iteratively:

$$\begin{cases} r_{merged,k}^{-} = r_{merged,k-1}^{-} + r_{i,j}^{-} - r_{i,j-1}^{-} \\ r_{merged,k}^{+} = r_{merged,k-1}^{-} + \varepsilon\sum_i N_i \end{cases}$$

Where $r_{i,j}$ is from the *j* th tuple in the *i* th local sketch

Global ε-approximate sketch

- Theorem 2: The merged sketch is ε/2-approximate
  - *For any tuple(v_i,r_i^-,r_i^+) in merged sketch, verify:*

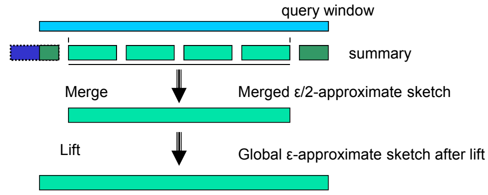
$$r_i^+ - r_{i-1}^- \le \varepsilon \sum_i N_i$$

$$r_i^- \le r_v \le r_i^+$$

Global ε-approximate sketch

- *Step 2:* lift the summary by εN/2

Lift operation: add εN/2 to each r_i^+



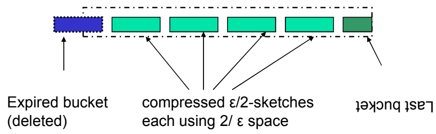
Global ε-approximate sketch

Theorem 3: Given an ε/2-approximate sketch on (1- ε/2)N data items, then lifting the sketch by εN/2 results in an ε-approximate sketch for the set of N data items.

Query the summary for any ϕ-quantile (first-hit):

$$\begin{cases} \lfloor \phi N \rfloor - r^- \le \varepsilon N \\ r^+ - \lfloor \phi N \rfloor \le \varepsilon N \end{cases}$$

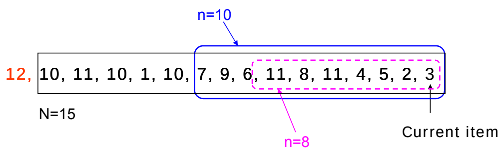
Space Complexity for sliding window



The total space needed is $O(\frac{1}{\varepsilon}\log \varepsilon^2 N + \frac{1}{\varepsilon^2})$

Variable length sliding window

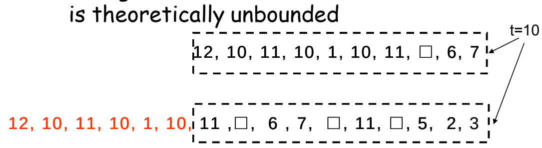
- n-of-N model :
  - Answer all sliding window queries with window length n (n≤N)



Other window semantics

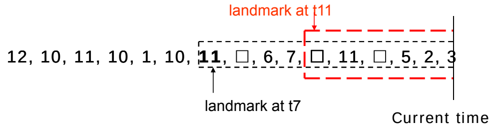
The sliding window based on a most recent time period

Challenge: Actual number of data elements is theoretically unbounded



Other window semantics

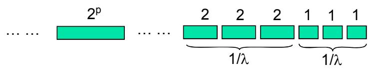
- Landmark windows



The Exponential Histogram (EH) by M.Datar et.al [DGIM02]

- In a λ-EH,

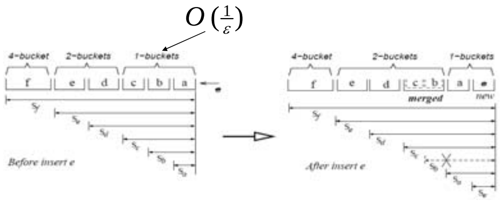
$$n_i \le \lambda \sum_{j=i+1}^{m} n_j$$



- $O(\frac{1}{\lambda}\log \lambda N)$ buckets for N data elements

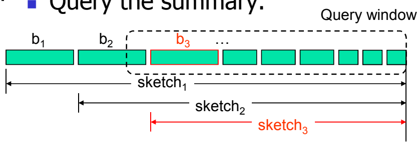
Quantile summary for n-of-N model (ICDE04, ours)

$O(\frac{1}{\varepsilon})$



Quantile summary for n-of-N query

- Query the summary.



Easy to extend to time window and landmark windows

Quantile summary for n-of-N model

- Outline of the Algorithm [icde04, unsw]

  Maintenance:
  - Partition a data stream by $\frac{\varepsilon}{2}$-EH (Exponential Histogram)
  - For each bucket, maintain an $\frac{\varepsilon}{2}$-approximate sketch to the current data item
  - Delete redundant buckets and expired buckets

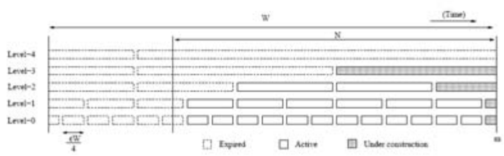
  Query:
  - Get one sketch to answer quantile query on most recent n items

  Space: $O(\max\{\frac{1}{\varepsilon^2}\log \varepsilon^2 N, \frac{1}{\varepsilon}\log^2 \varepsilon^2 N\})$

More result [PODS04, Stanford]

sliding window: $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\log N)$ ($O(\max\{\frac{1}{\varepsilon}\log \varepsilon^2 N, \frac{1}{\varepsilon^2}\})$)

n-of-N: $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\log \varepsilon N \log N)$



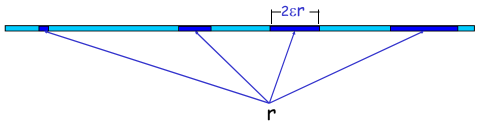
Relative Error Techniques

- Relative Є-approximate

Given r, return any element e with rank r' such that

$\frac{|r-r'|}{r} \leq \varepsilon$

Space Lower bound : O(log(εn)/ε)



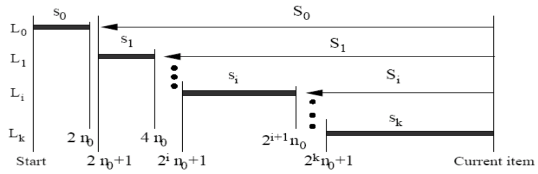
Applications

- Skewed data. Like IP network traffic data
  - Long tails of great interest
  - Eg: 0.9, 0.95, 0.99-quantiles of TCP round trip times
- In some applications, "head" or "tail" is the most important part.
- Counting inversions
- etc.

Existing Techniques

- GZ Algorithm [SODA03, Bell Lab]
  - Space $O(1/\varepsilon^3 \log N)$, need to know N in advance
- CKMS Algorithm [ICDE05, AT & T]
  - No sub-linear space bound guarantee
  - Extend GK-algorithm

MR [icde06, UNSW]



$R_0 = [1, 2n_0]$ $R_i = [2^i n_0, 2^{i+1} n_0]$ Sampling rate : 2^i

L_i become active when N = $2^i n_0 + 1$

s_i samples over first $2^i n_0$ elements, will not change later

S_i samples over other elements, keep at most n_0 smallest samples

MR - Correctness

For the query $r \in [2^i n_0, 2^{i+1} n_0]$

- | r - r(e) |<= $\varepsilon 2^i n_0$ ⟹ | r - r(e) |<= ε r

How many samples is required for each sample set (s_i, S_i) ?

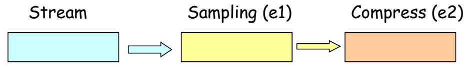
n_0 >= O($1/\varepsilon^2$ log($1/\delta$))

MR [icde06, UNSW]

- Without priori knowledge of N , with probability at lest $1-\delta$, we can get the relative ε- approximate quantile with space $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\log\varepsilon^2 N)$
- Processing time per element $O(\log(\frac{\epsilon^2 N}{\log 1/\delta})\log(\epsilon^{-2}\log\frac{1}{\delta}))$
- Query time $O(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\log\log(\frac{\epsilon^2 N}{\log 1/\delta}))$

MRC [ICDE06, UNSW]

- Feed samples to compress algorithm (GK)



Pipeline

Space bound

Average case $O(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon}\log\frac{1}{\delta})\frac{\log^{2+\alpha}\varepsilon N}{1-1/2^{\alpha}})$ for an $\alpha > 0$

Worst case $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\log\varepsilon^2 N)$

More results [PODS06, AT & T]

Deterministic algorithm is proposed for fixed value domain

Space bound $O(\frac{\log|U|}{\varepsilon}\log(\varepsilon N))$

The problem of sliding window is not well solved…

Duplicate-insensitive Technique

Given : a set of data elements S={(x, v)} where x is the element and v=f(x).

- Elements are sorted on a monotonic order of v.
- Duplicates may exist.
- D_S: set of distinct elements in S.
- Rank Queries (quantiles) are against D_S

Example

Data Stream:
(x_1, 1) , (x_5,6) , (x_1,1) , (x_2,1) , (x_4,10) , (x_2,1) , (x_3,7) , (x_4,10)

Sorted Distinct Sequence:

(x_1, 1) , (x_2,1) , (x_5,6) , (x_3,7) , (x_4,10)

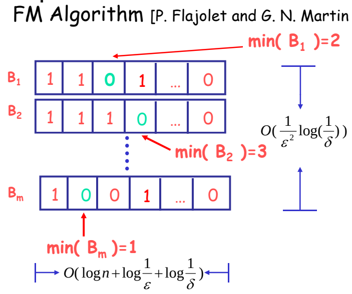
r=3
(0.5-quantile)

Applications

- Projections
- IP network monitoring
- Sensor network
- etc

Preliminaries

FM Algorithm [P. Flajolet and G. N. Martin , FOCS83]



$P(h(x) = i) = \frac{1}{2^{i+1}}$

$A = \frac{1}{\varphi} 2^{\sum_{i=1}^{m} \min(B_i)/m}$

$\varphi \approx 0.775351$

Important properties :

$fm(P \cup Q) = fm(P) \vee fm(Q).$

With confidence 1-δ, count (1-ε) < A < count (1+ε).

Uniform Error technique

[Pods 06, Bell Lab & Rugters Uni]

Distinct Range Sum: Count-Min + FM

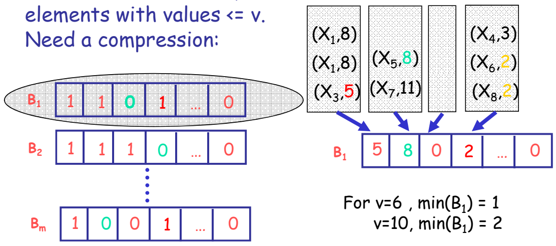
Space $O(\frac{1}{\epsilon^3}\log\frac{1}{\delta}\log^2 m)$

[SIGMOD05, UCSB&Intel] & [Tech Report06, Boston]

Apply FM; Space: $O(\frac{1}{\epsilon^3}\log\frac{1}{\delta}\log m)$

Relative Error technique [ICDE 07, UNSW]

Basic Idea: for each v, build FM Sketch for elements with values <= v.
Need a compression:



For v=6 , min(B_1) = 1
v=10, min(B_1) = 2

[ICDE07, UNSW]

- ϵ-Approximate with confidence 1 - δ,
- space: $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\log N)$
- various ways to speed up the algorithm

Miscellaneous

- Continuous Queries
  - continuous monitor the network [sigmod06, Bell Lab]
  - Massive set of rank queries [TKDE06, UNSW]
- Quantile computation against high dimensional data
  - R tree based algorithm. [EDBT06, CUHK]
  - Adaptive partition algorithm. [ISAAC 04, UCSB]

Open Problems

- Uncertainty data
  Challenge : the value of the element is not fixed!
- Graphs
  - common to model real applications
    IP network, communication network, WWW, etc
  - summarize distribution of various node degree information

  Challenge : the graph structure is continuously "disclosed "!

Reference

- **[sigmod01, PSU]** M. Greenwald and S. Khanna. "Space-efficient online computation of quantile summaries" . In SIGMOD 2001.
- **[Sigmod99, IBM]** G. S. Manku, S. Rajagopalan, and B. G. Lindsay. "Random sampling techniques for space efficient online computation of order statistics of large datasets". In SIGMOD 1999.
- **[LATIN04, Rutgers Uni]** G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. In LATIN 2004.
- **[icde04, UNSW]** X. LIN, H. Lu, J. Xu, and J.X. Yu, "Continuously Maintaining Quantile Summaries of the Most Recent N Elements over a Data Stream", In ICDE2004.
- **[DGIM02]** Mayur Datar, Aristides Gionis, Piotr Indyk, Rajeev Motwani "Maintaining stream statistics over sliding windows (extended abstract)" , In SODA 2002
- **[PODS04, Stanford]** A Arasu and G S Manku, "Approximate Frequency Counts over Data Streams", In PODS 2004.
- **[SODA03, Bell Lab]** A. Gupta and F. Zane. "Counting inversions in lists". In SODA 2003.
- **[ICDE05, AT&T] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. "Effective computation of biased quantiles over data streams" In ICDE 2005.**
- **[icde06, UNSW]** Y. Zhang, X. LIN, J. Xu, F. Korn, W. Wang, "Space-efficient Relative Error Order Sketch over Data Streams", ICDE 2006.

Reference

- **[PODS06, AT&T]** G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava. "Space- and time-efficient deterministic algorithms for biased quantiles over data streams", In PODS 2006.
- **[Pods 05, Bell Lab & Rugters Uni]** G. Cormode and S. Muthukrishnan. "Space efficient mining of multigraph streams", In PODS 2005.
- **[SIGMOD05, UCSB&Intel]** A. Manjhi, S. Nath, and P. B. Gibbons. "Tributaries and deltas: Efficient and robust aggregation in sensor network streams" In SIGMOD 2005.
- **[Tech Report05, Boston]** M. Hadjieleftheriou, J.W. Byers, and G. Kollios "Robust sketching and aggregation of distributed data streams" , Technical report, Boston University, 2005.
- **[ICDE 07, UNSW]** Y. Zhang, X. Lin, Y. Yuan, M. Kitsuregawa, X. Zhou, and J. Yu. "Summarizing order statistics over data streams with duplicates"(poster) In ICDE 2007.
- **[sigmod06, Bell Lab]** G. Cormode, R. Keralapura, and J. Ramimirtham. Communication-efficient distributed monitoring of thresholded counts. In SIGMOD, 2006.
- **[TKDE06, UNSW]** X. Lin, J. Xu, Q. Zhang, H. Lu, J. Yu, X. Zhou, and Y. Yuan. "Approximate processing of massive continuous quantile queries over high speed data streams", TKDE 2006.
- **[EDBT06, CUHK]** M. Yiu, N. Marmoulis, and Y. Tao. "Efficient quantile retrieval on multi-dimensional data". In EDBT 2006.
- **[ISAAC 04, UCSB]** J. Hershberger, N. Shrivastava, S. Suri, and C. Toth. "Adaptive spatial partitioning for multidimensional data streams" , In ISAAC 2004.
- **[P. Flajolet and G. N. Martin,FOCS83]** P.Flajolet,G.Nigel Martin "Probabilistic Counting " in FOCS 1983