Rectangle-Efficient Aggregation in Spatial Data Streams

Srikanta Tirthapura
Iowa State

David Woodruff
IBM Almaden

The Data Stream Model

- Stream S of additive updates (i, Δ) to an underlying vector v:

$$v_i \leftarrow v_i + \Delta$$

- Dimension of v is n
- v initialized to 0^n
- Number of updates is m
- Δ is an integer in $\{-M, -M+1, \ldots, M\}$
- Assume $M, m \leq \mathrm{poly}(n)$

Applications

- Coordinates of v associated with items
- v_i is the number of times (frequency) item i occurs in S
- Number of non-zero entries of v = number of distinct items in S
  - denoted $|v|_0$
- $|v|_1 = \Sigma_i |v_i|$ is sum of frequencies in S
- $|v|_2^2 = \Sigma_i v_i^2$ is self-join size
- $|v|_p = (\Sigma_i v_i^p)^{1/p}$ is p-norm of v

Lots of Known Results

- (ε, δ)-approximation
  - output estimator E to $|v|_p$
  - $\Pr[|v|_p \leq E \leq (1+\varepsilon)\, |v|_p] \geq 1-\delta$

- Let O~(1) denote poly($1/\varepsilon$, $\log n/\delta$)

- Optimal estimators for $|v|_p$:
  - $0 \leq p \leq 2$, use O~(1) memory
  - $p > 2$ use O~($n^{1-2/p}$) memory
  - Both results have O~(1) update time

Range-Updates

- Sometimes more efficient to represent additive updates in the form ([i,j], Δ):

  $\forall k \in \{i, i+1, i+2, \ldots, j\}$: v_k <- $v_k + \Delta$

- Useful for representing updates to time intervals

- Many reductions:
  - Triangle counting
  - Distinct Summation
  - Max Dominance Norm

A Trivial Solution?

- Given update ([i, j], Δ), treat it as j-i+1 updates (i, Δ), (i+1, Δ), (i+2, Δ), …, (i+j, Δ)

- Run memory-optimal algorithm on the resulting j-i+1 updates

- Main problem: update time could be as large as n

Scattered Known Results

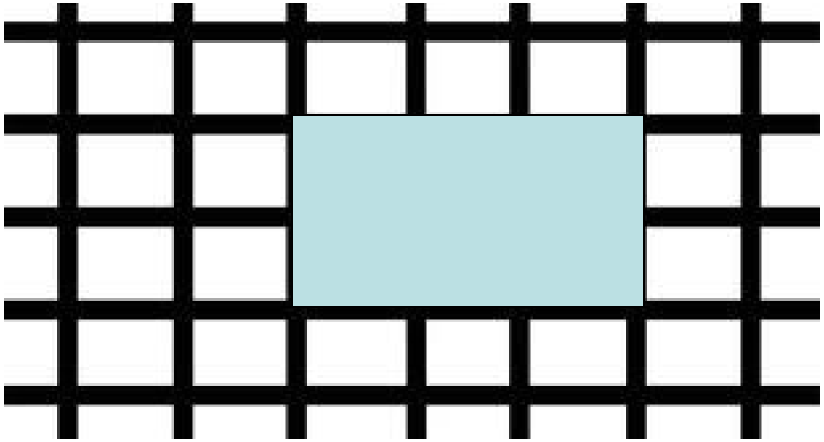
- Algorithm is range-efficient if update time is O~(1) and memory optimal up to O~(1)
- Range-efficient algorithms exist for
  - positive-only updates to $|v|_0$
  - $|v|_2$
- For $|v|_p$, $p>2$, can get $O\sim(n^{1-1/p})$ time and memory
- Many questions open (we will resolve some)

Talk Outline

- General Framework: Rectangle Updates
- Problems
  - Frequent points (“heavy hitters”)
  - Norm estimation
- Results
- Techniques
- Conclusion

From Ranges to Rectangles

Vector v has coordinates indexed by pairs $(i,j) \in \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$

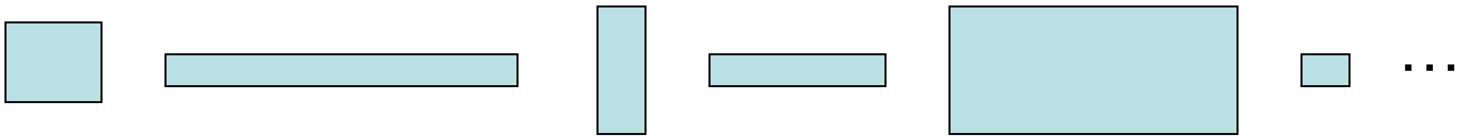


- Rectangular updates $([i_1, j_1] \times [i_2, j_2], \Delta)$

$$\forall (i,j) \in [i_1, j_1] \times [i_2, j_2]: v_{i,j} \leftarrow v_{i,j} + \Delta$$

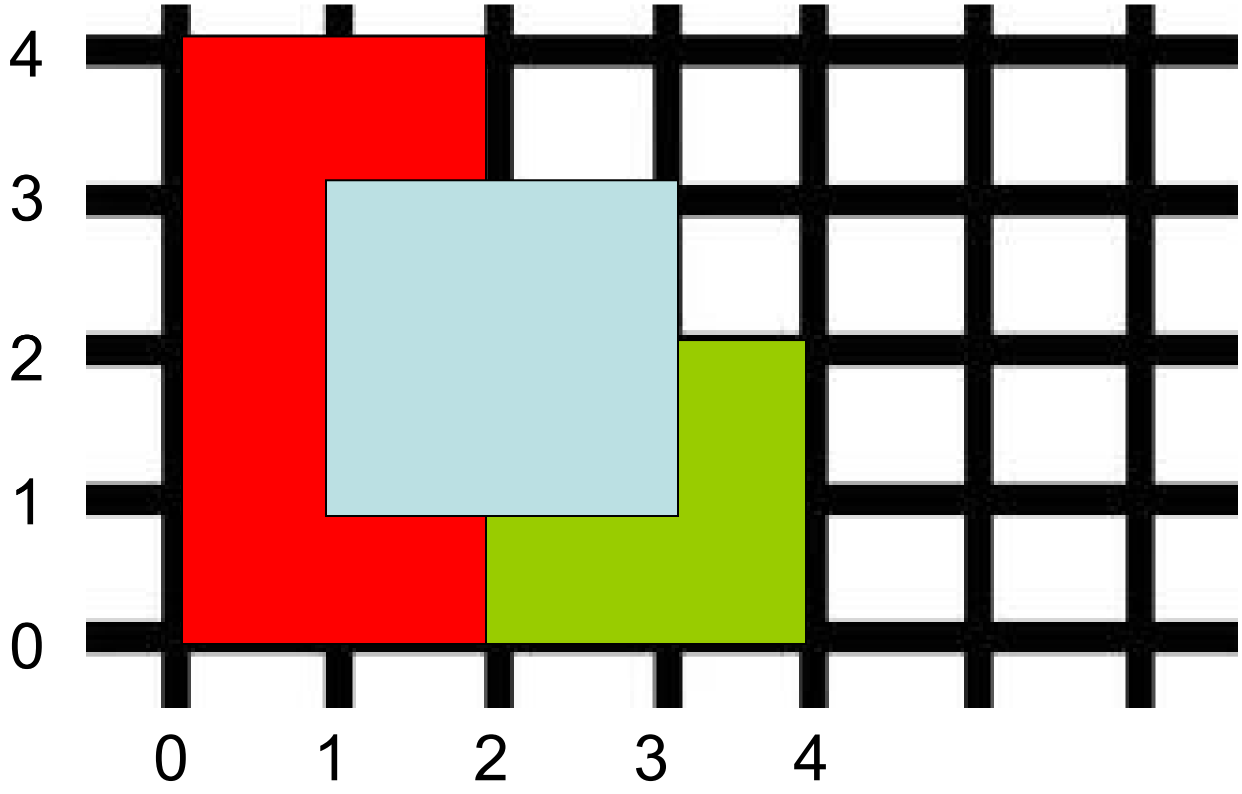
Spatial Datasets

- A natural extension of 1-dimensional ranges is to axis-aligned rectangles
- Can approximate polygons by rectangles
- Spatial databases such as OpenGIS



- No previous work on streams of rectangles
- What statistics do we care about?

Klee's Measure Problem



What is the volume of the union of three rectangles?

$[0, 2] \times [0,4]$, $[1, 3] \times [1,3]$, $[2, 4] \times [0,2]$

Other Important Statistics

- $\max_{i,j} v_{i,j}$ is the depth
- can approximate the depth by approximating $|v|_p$ for large $p > 0$ (sometimes easier than computing depth)
- also of interest: find "heavy" hitters, namely, those (i,j) with $v_{i,j} \geq \varepsilon\ |v|_1$ or $v_{i,j}^2 \geq \varepsilon\ |v|_2^2$

Notation

- Algorithm is rectangle-efficient if update time is O~(1) and memory optimal up to O~(1)

- For (ε,δ)-approximating $|v|_p$ we want
  - for $0 \leq p \leq 2$, O~(1) time and memory
  - for $p > 2$, O~(1) time and $O\sim(n^2)^{1-2/p}$ memory

Our Results

- For $|v|_p$ for $0 \leq p \leq 2$, we obtain the first rectangle-efficient algorithms
  - First solution for Klee's Measure Problem on streams
- For finding heavy hitters, we obtain the first rectangle-efficient algorithms
- For $p > 2$ we achieve:
  - $\tilde{O}(n^2)^{1-2/p}$ memory and time

    OR
  - $\tilde{O}(n^2)^{1-1/p}$ memory and $\tilde{O}(1)$ time

Our Results

- For any number d of dimensions:

- For $|v|_p$ for $0 \leq p \leq 2$ and heavy hitters:
  - O~(1) memory and O~(d) time

- For $|v|_p$ for $p > 2$:
  - $O\sim(n^d)^{1-2/p}$ memory and time

    OR
  - $O\sim(n^d)^{1-1/p}$ memory and O~(d) time

- Only a mild dependence on d
- Improves previous results even for d = 1

Our Techniques

- Main idea:
  - *Leverage a technique in streaming algorithms for estimating* $|v|_p$ *for any* $p \geq 0$
  - *Replace random hash functions in technique with hash functions with very special properties*

Indyk/W Methodology

- To estimate $|v|_p$ of a vector v of dimension n:

- Choose O(log n) random subsets of coordinates of v, denoted $S^0, S^1, \ldots, S^{\log n}$
  - S^i is of size $n/2^i$

- $\forall S^i$: find those coordinates $j \in S^i$ for which $v_j^2 \geq \gamma \cdot |v_{S^i}|_2^2$
  - Use CountSketch:
  - Assign each coordinate j a random sign $\sigma(j) \in \{-1,1\}$
  - Randomly hash the coordinates into $1/\gamma$ buckets, maintain $\Sigma_{j \text{ s.t. } h(j) = k \text{ and } j \in S^i} \sigma(j) \cdot v_j$ in k-th bucket

	$\Sigma_{j \text{ s.t. } h(j) = 2 \text{ and } j \in S^i} \ \sigma(j) \cdot v_j$		

- Our observation: can choose the S^i and hash functions in CountSketch to be pairwise-independent

Special Hash Functions

- Let A be a random k x r binary matrix, and b a random binary vector of length k

- Let x in $GF(2^r)$

- Ax + b is a pairwise-independent function:
  – For $x \neq x' \in GF(2^r)$ and $y \neq y' \in GF(2^k)$:

$$\Pr[Ax+b = y \text{ and } Ax'+b = y'] = 1/2^{2k}$$

Special Hash Functions Con'd

- Given a stream update ($[i_1, j_1] \times [i_2, j_2], \Delta$):
  can decompose $[i_1, j_1]$ and $[i_2, j_2]$ into O(log n) disjoint *dyadic intervals:*

  $$[i_1, j_1] = [a_1, b_1] \cup \ldots \cup [a_s, b_s]$$

  $$[i_2, j_2] = [c_1, d_1] \cup \ldots \cup [c_{s'}, d_{s'}]$$

- Dyadic inteval: $[u2^q, (u+1)2^q)$ for integers u,q

- Then $[i_1, j_1] \times [i_2, j_2] = \cup_{r, r'} [a_r, b_r] \times [c_{r'}, d_{r'}]$

Special Hash Functions Con'd

- A property of function $Ax+b$: can quickly compute the number of x in the interval $[a_r, b_r] \times [c_{r'}, d_{r'}]$ for which $Ax+b = e$

- By the structure of dyadic intervals, this corresponds to fixing a subset of bits of x, and letting the remaining variables be free:

  # of $x \in [a_r,b_r] \times [c_{r'},d_{r'}]$ for which $Ax+b = e$ is

  # of z for which $A'z = e'$, z is unconstrained

Can now use Gaussian elimination to count # of z

Techniques WrapUp

- Step 1: Modify Indyk/W analysis to use only pairwise-independence
- Step 2: Given update $([i_1, j_1] \times [i_2, j_2], \Delta)$, decompose into disjoint products of dyadic intervals $\cup_{r, r'} [a_r, b_r] \times [c_{r'}, d_{r'}]$
- Step 3: For each $[a_r, b_r] \times [c_{r'}, d_{r'}]$, find the number of items in each S^i and each bucket of CountSketch and with each sign by solving a system of linear equations
- Step 4: Update all buckets

Conclusion

Contributions:

- Initiated study of rectangle-efficiency
- Gave rectangle-efficient algorithms for estimating $|v|_p$, $0 \leq p \leq 2$ and heavy hitters
- Tradeoffs for estimating $|v|_p$ for $p > 2$
- Improve previous work even for $d = 1$

Open Questions:

- Get $\tilde{O}((n^d)^{1-2/p})$ memory and $\tilde{O}(1)$ time for $p > 2$
- Other applications of the model and technique