At-the-time and Back-in-time Persistent Sketches

Benwei Shi, Zhuoyue Zhao, Yanqing Peng, Feifei Li, Jeff M. Phillips

THE UNIVERSITY OF UTAH®

Use case: website access log

ip	date	time	other
67.85.119.gii	1/1/2021	00:00:04	…
216.202.166.fie	1/1/2021	00:00:07	…
67.85.119.gii	1/1/2021	00:00:07	…
208.191.58.ced	1/1/2021	00:00:10	…
216.202.166.fie	1/1/2021	00:00:11	…
216.243.8.afh	1/1/2021	00:00:17	…
67.85.119.gii	1/1/2021	00:00:19	…
216.191.239.jdf	1/1/2021	00:00:23	…
216.191.239.jdf	1/1/2021	00:00:24	…
…	…	…	…
64.12.96.cib	5/25/2021	23:59:59	…

- Frequent IPs from beginning to now?
- Frequent IPs from beginning to time t?
- Frequent IPs from time t to now?

Settings and notations

Given a data stream $A = (a_0, a_1, \cdots, a_{n-1})$, with length n, too big to store locally.

Data:	a_0	a_1	$\cdots$	a_{t-1}	a_t	$\cdots$	a_{n-t}	$\cdots$	a_{n-2}	a_{n-1}
A	0									$n-1$
A^t	0			$t-1$						
A^{-t}							$n-t$			$n-1$

- *Streaming sketch* for A, only the last state of the whole stream.
- *At-the-time persistent (ATTP) sketch for* A^t, first t items in the stream, for any $0 \leq t \leq n$ given later.
  - Auditing any prior states of the stream
- *Back-in-time persistent (BITP) sketch* for A^{-t}, last t items in the stream, for any $0 \leq t \leq n$ given later.
  - Like sliding windows with all possible window size t.

Example: Streaming Reservoir Sampling

Algorithm: Reservoir Sampling

Input: $A = (a_0, a_1, \cdots, a_{n-1})$

for $i = 0, \cdots, n-1$ **do**

  $b \leftarrow a_i$ with probability $1/(i+1)$

end for

return b

---

i	0	1	2	3	4	5	6	7	8	9	10
a_i	z	x	y	x	x	y	x	z	y	x	x
Streaming sketch	~~$b =$ z~~	~~$b =$ x~~				~~$b =$ y~~				$b =$ x	

Example: ATTP Reservoir Sampling

Algorithm: ATTP Reservoir Sampling

Input: $A = (a_0, a_1, \cdots, a_{n-1}), j \leftarrow 0$

for $i = 0, \cdots, n-1$ **do**

$b_j \leftarrow a_i, t_j \leftarrow j, j \leftarrow j + 1$ with probability $1/(i+1)$

end for

return $b_0, \cdots, b_{j-1}, t_0, \cdots, t_{j-1}$

i	0	1	2	3	4	5	6	7	8	9	10
a_i	z	x	y	x	x	y	x	z	y	x	x
ATTP sketch	b_0 =z t_0 =0	b_1 =x t_1 =1				b_2 =y t_2 =5				b_3 =x t_3 =9	

- Simply run k ATTP Reservoir Sampling simultaneously to get a ATTP version of k random samples with replacement.

Random Sampling: Streaming vs ATTP

- Above two algorithm share same running time.
- ATTP sketch need more space for the sketch history.
  - Let S_n be the size of the ATTP Reservoir Sampling, we show that

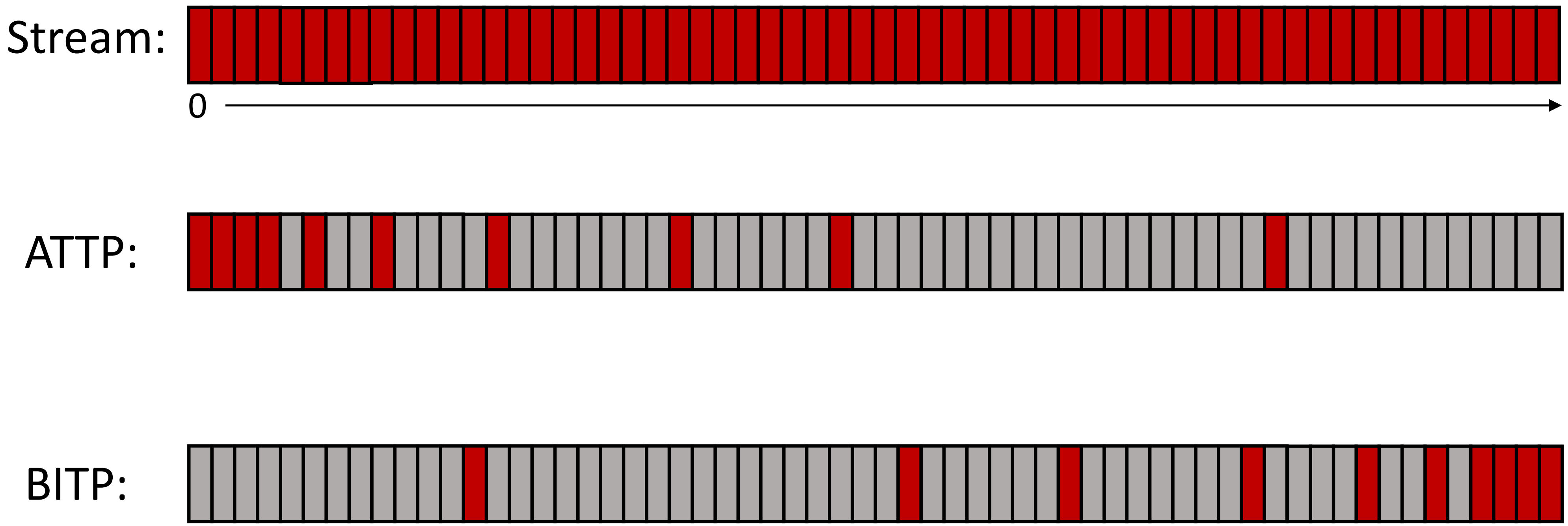
$$\mathbb{E}(S_n) \leq \ln n + 1 \in O(\ln n)$$

- A random sample is one of the most versatile data sketch

Problems	**Streaming (k)**	**ATTP**
ε-QuantilesEstimation	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{n}{\delta}\right)$
ε-FrequencyEstimation	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{n}{\delta}\right)$
ε-ApproximateRangeCount with VC-dimension ν	$O\left(\frac{\nu}{\varepsilon^2}\log\frac{1}{\delta}\right)$	$O\left(\frac{\nu}{\varepsilon^2}\log\frac{n}{\delta}\right)$
ε-KernelDensityEstimation	$O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$	$O\left(\frac{1}{\varepsilon^2}\log\frac{n}{\delta}\right)$

ATTP Random Sampling vs BITP Random Sampling

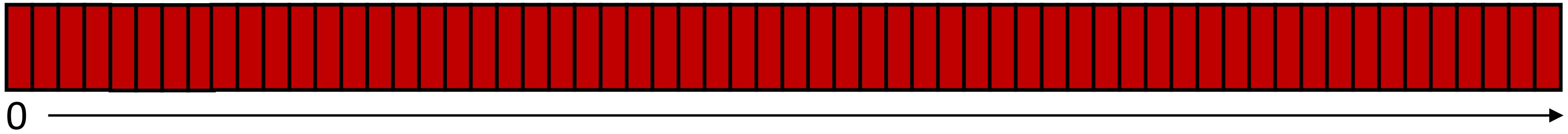
- Without replacement sampling of size $k = 4$



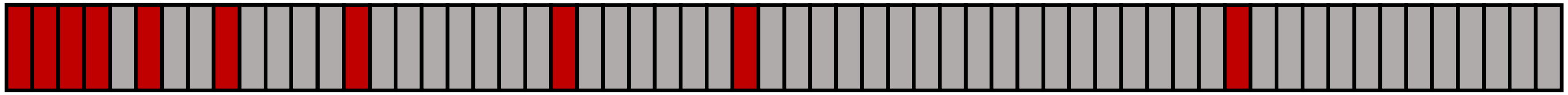
ATTP Random Sampling vs BITP Random Sampling

- Without replacement sampling of size $k = 4$

Stream:

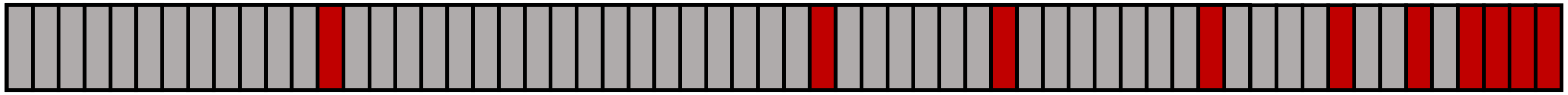


ATTP:



Space: $O(k \log n)$, update time:$O(1)$

BITP:

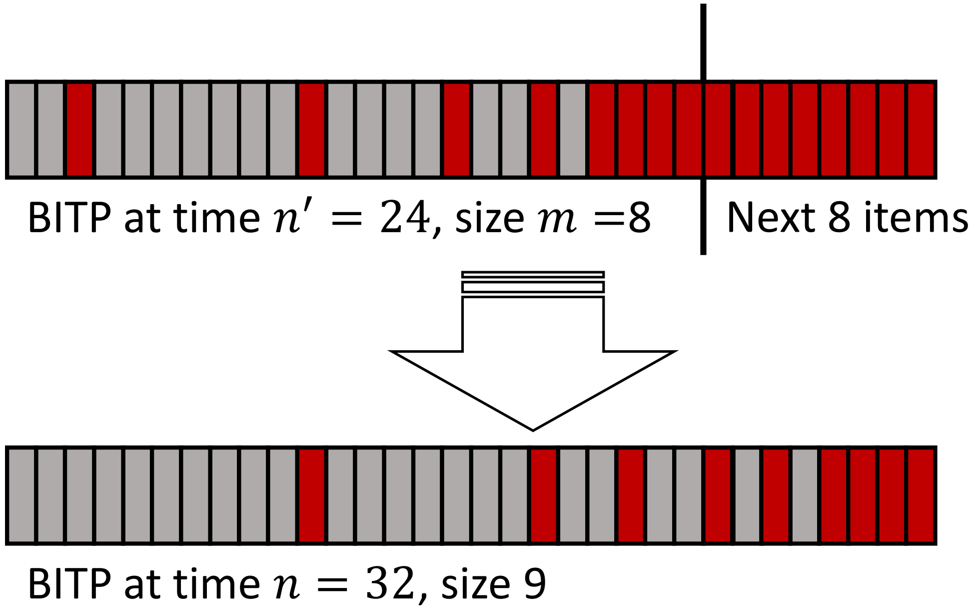


Space: $O(k \log n)$, update time:$O(k \log n)$

Can we do better?

BITP Random Sampling

- Without replacement sampling of size $k = 4$



- Same size with ATTP random sample asymptoticly.
- Same update time, $O(1)$, for each item asymptoticly.
- If using interval tree to speed up query time, then need $O(\log k)$ update time.

Streaming Misra-Gries

i: 0 1 2 3 4 5 6 7 8 9 10

a_i: 

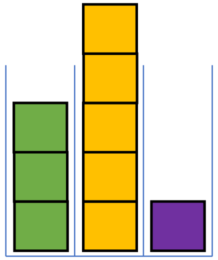
Frequencies:

$f($$) = 3$

$f($$) = 6$

$f($$) = 1$

$f($$) = 1$



Estimations:

$\hat{f}($$) = 2$

$\hat{f}($$) = 5$

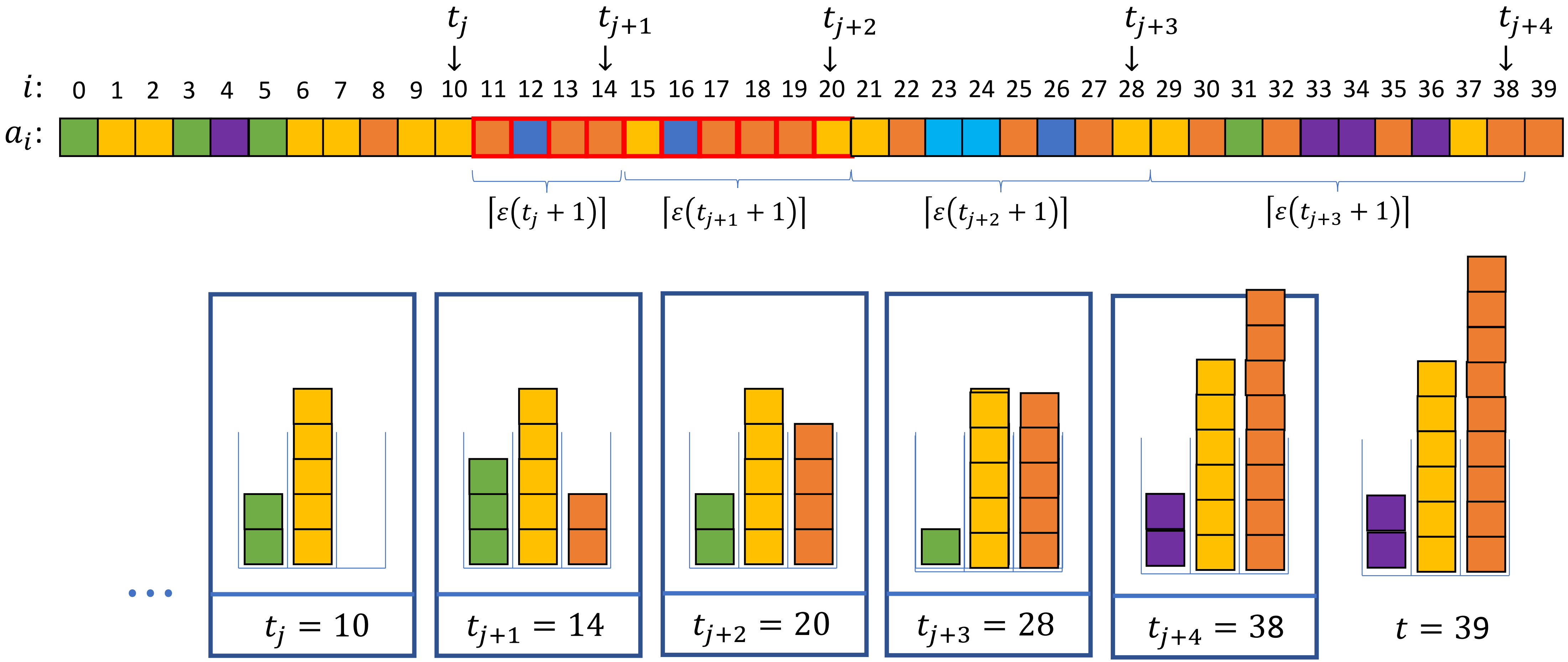
$\hat{f}($$) = 0$

$\hat{f}($$) = 0$

Space: $\frac{1}{\varepsilon} = 3$,

An ε-FrequencyEstimation: $f(a) - \hat{f}(a) \le \varepsilon n$

ATTP Misra-Gries with Checkpoints

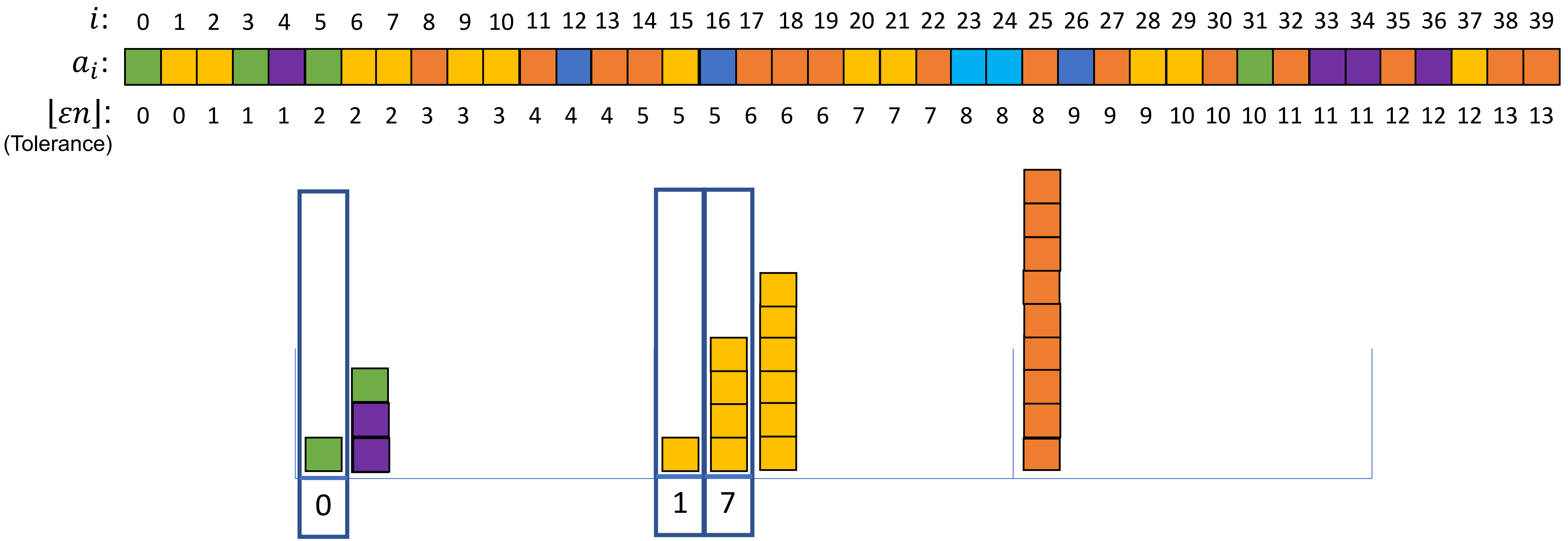


An ATTP (2ε)-FrequencyEstimation: $f(a) - \hat{f}(a) \leq 2\varepsilon n$

need space $O\left(\frac{1}{\varepsilon^2} \log n\right)$ at most.

ATTP Misra-Gries with Elementwise Improvements

Chain Misra-Gries (CMG)



An ATTP (2ε)-FrequencyEstimation: $f(a) - \hat{f}(a) \leq 2\varepsilon n$

need space $O\left(\frac{1}{\varepsilon} \log n\right)$ at most.

From Mergeability to ATTP and BITP Sketches

- Any “mergeable” (includes sampling, linear) sketches can be made ATTP and BITP with small overhead.

- Some more specialized ATTP and BITP results:

Algorithms	For Problem	ATTP Size	BITP Size
Misra-Gries	ε-FrequencyEstimation	$\mathbf{O}\left(\frac{\mathbf{1}}{\boldsymbol{\varepsilon}}\log \mathbf{n}\right)$ (Chain Misra-Gries)	$\mathbf{O}\left(\frac{\mathbf{1}}{\boldsymbol{\varepsilon}^2}\log \mathbf{n}\right)$ (Tree Misra-Gries)
FrequentDirections	ε-MatrixCovariance	$\mathbf{O}\left(\frac{\mathbf{d}}{\boldsymbol{\varepsilon}}\log \lVert\mathbf{A}\rVert_{\mathbf{F}}\right)$	$\mathbf{O}\left(\frac{\mathbf{1}}{\boldsymbol{\varepsilon}^2}\log \lVert\mathbf{A}\rVert_{\mathbf{F}}\right)$

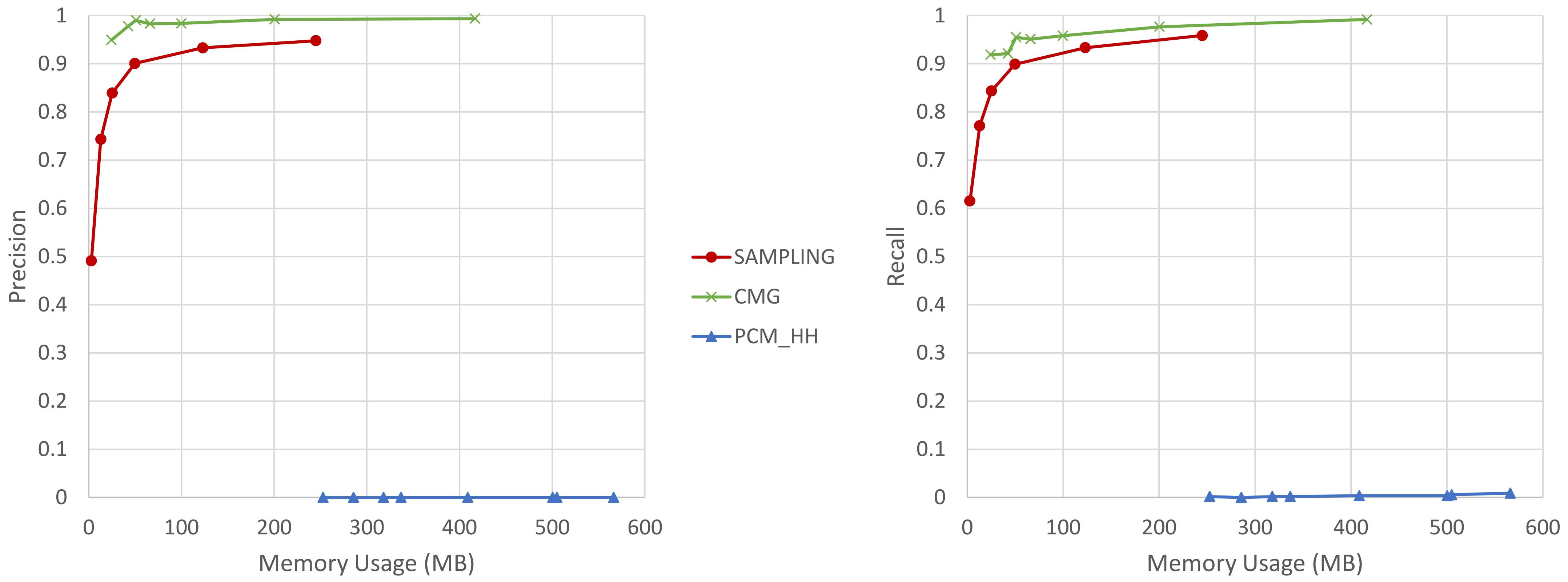
Main results for ATTP and BITP sketches

Problems	BITP	ATTP
ε-QuantilesEstimation	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$
wt. ε-QuantilesEstimation	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$
ε-FrequencyEstimation	$O\left(\left(1/\varepsilon^2\right)\log n\right)$	$O((1/\varepsilon)\log n)$
wt. ε-FrequentEstimation	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$
ε-ApproximateRangeCount	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$
wt. ε-ApproximateRangeCount	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$
ε-KernelDensityEstimation	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$	$O\left(\left(1/\varepsilon^2\right)\log n\right)^*$
ε-MatrixCovariance	$O\left(\left(d/\varepsilon^2\right)\log\|A\|_F\right)$	$O((d/\varepsilon)\log\|A\|_F)$
(α,ε)-MatrixCovariance		$O\left(\left(d^2/\alpha\right)\log d\log\left((\alpha/\varepsilon)\|A\|_2^2\right)\right)$

Weighted (wt.) with U-bounded weights: $\frac{\max weight}{\min weight} \le U$, and $U = poly(n)$;

*: Randomized with high probability.

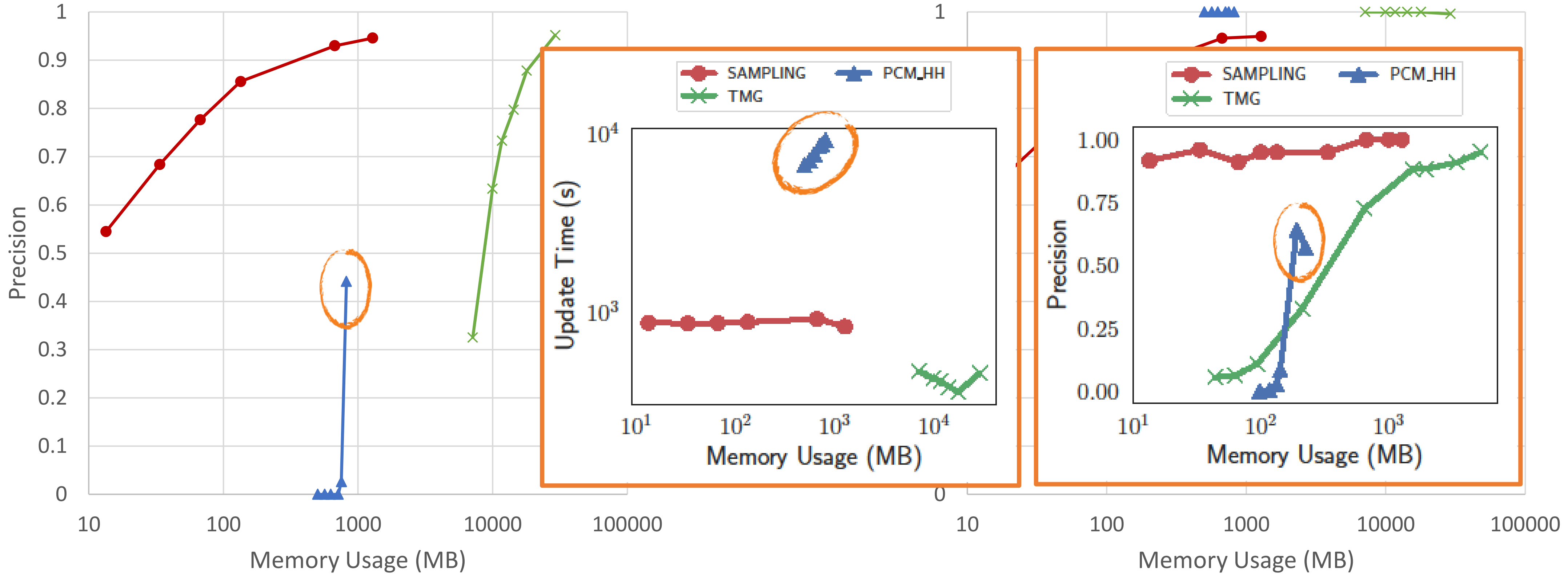
Experiments: ATTP Heavy Hitter



- Sampling: ATTP random sampling without replacement.
- CMG: ATTP Chain Misra-Gries.
- PCMHH: persistent Count-Min sketch (Z. Wei et. al., 2015).

Z. Wei et. al.. Persistent data sketching. In SIGMOD, 2015.

Experiments: BITP Heavy Hitter



- Sampling: BITP random sampling without replacement.
- TMG: Tree Misra-Gries.
- PCMHH: persistent Count-Min sketch.

Conclusion

- We defined the new concept of ATTP sketches and BITP sketches which only require a logarithmic overhead on most existing sketches.
- We described frameworks for making ATTP/BITP sketches from existing streaming/mergeable sketches. We also gave several ATTP and BITP algorithms and theoretically analyzed them for different types of sketching problems.

Data	a_0	a_1	$\cdots$	a_{t-1}	a_t	$\cdots$	a_{n-t}	$\cdots$	a_{n-2}	a_{n-1}
A	0									$n-1$
A^t	0			$t-1$						
A^{-t}							$n-t$			$n-1$

- A *streaming sketch* for A.
- An *ATTP sketch* for A^t for any $0 \leq t \leq n$ given later.
- A *BITP sketch* for A^{-t} for any $0 \leq t \leq n$ given later.