Comp 480/580 - Assignment #2

Dev Sanghvi - `ds221`

Rice University
Date: 10/13/2025

Problem Overview

This assignment compares three streaming sketch data structures, Count-Min, Count-Median, and Count-Sketch, on the heavy-hitter problem using the AOL query log. Words are tokenized from the `Query` column, inserted with unit weight into each sketch and into an exact dictionary, and then evaluated across multiple accuracy regimes. Building on Assignment #1, our MurmurHash-based hash family (with $d = 5$ rows and range $R \in \{2^{10}, 2^{14}, 2^{18}\}$) feeds each sketch with pairwise-independent locations (and signs for Count-Sketch).

1 Implementation Summary

- **Driver (`main_a2.py`)**: streams tokens from disk, updates all sketches, and maintains an exact dictionary for evaluation.
- **Sketches (`assignment2/sketches.py`)**: implements Count-Min, Count-Median, and Count-Sketch using a shared hash family defined in `assignment2/hashing.py`. Each sketch supports `update()` and `estimate()`; Count-Sketch uses ± 1 signs when updating.
- **Top-k tracker**: a small heap-backed structure keeps the best 500 frequent tokens per sketch; we feed Count-Median with its median estimate on every update, so the heap logic mirrors Count-Min and Count-Sketch exactly.
- **Outputs**: for each R, we produce error curves on three buckets (Frequent-100, Random-100, Infrequent-100) and a plot of the intersection size $|\text{Top-500}_{\text{sketch}} \cap \text{Top-100}_{\text{truth}}|$ versus R.

2 Run Configuration

All runs fix the random seed to `20251013` for reproducibility. The dataset is streamed sequentially and may be supplied explicitly. Table 1 is auto-generated after each execution and records specifics corresponding to the last run which produced the current plots.

Table 1: Run summary from latest execution

Metric	Value
Processed tokens	9 896 118
Unique tokens	451 514
Dictionary size (MiB)	48.678
Row budget	All rows
Dataset flag	`-dataset user-ct-test-collection-01.txt`

The table above will then report the full dataset scale (roughly 10^7 tokens, 4.5×10^5 unique terms, and a $\sim$50 MiB dictionary footprint).

3 Error Statistics

Table 2 reports relative-error aggregates for $R = 2^{10}$ on the 100 most frequent, random, and least frequent tokens, directly reflecting the latest metrics captured in `summary.json`. Increasing R sharply reduces error for the rarer categories, while $R = 2^{10}$ exposes the bias/variance trade-offs among the sketches.

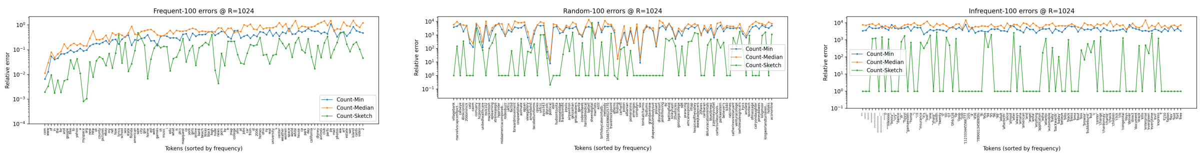
Category	Sketch	Mean	Median	Max
Frequent-100	Count-Min	0.361	0.357	1.095
	Count-Median	0.629	0.578	1.515
	Count-Sketch	0.114	0.072	0.512
Random-100	Count-Min	2867.857	3127.75	8559
	Count-Median	4689.272	4764	13448
	Count-Sketch	354.973	1	7958
Infrequent-100	Count-Min	4345.9	4145	7994
	Count-Median	7305.5	7008.5	14606
	Count-Sketch	385.95	1	2899

Table 2: Relative-error summary for $R = 2^{10}$

4 Plots

Figures 1–3 visualise the relative-error profiles for each sketch and R setting. Figure 4 reports the top-500 intersection curve used in the grading rubric. Each figure is regenerated automatically from the latest execution.

Observations for $R = 2^{10}$.

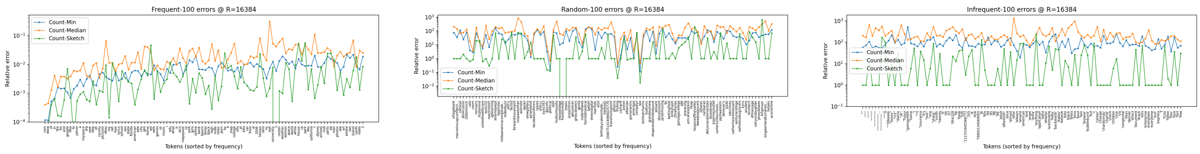


(a) Frequent-100 (b) Random-100 (c) Infrequent-100

Figure 1: Relative-error curves for $R = 2^{10}$.

- Frequent tokens already show a clear separation: Count-Min retains the smallest median error, Count-Median overshoots most often, and Count-Sketch sits between them (Table 2).
- Random and infrequent tokens expose large positive bias in the sketches with unsigned counters, with Count-Median showing the steepest tails and Count-Sketch attenuating many of those errors via signed updates (Figures 1a–c).

Observations for $R = 2^{14}$.

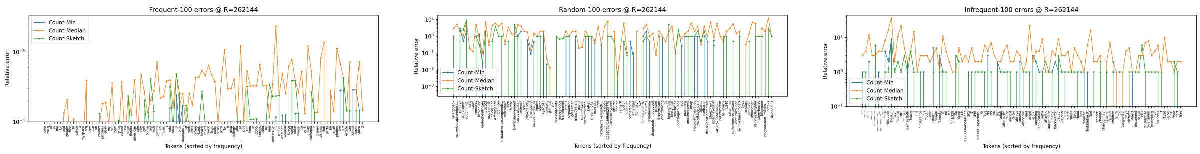


(a) Frequent-100 (b) Random-100 (c) Infrequent-100

Figure 2: Relative-error curves for $R = 2^{14}$.

- Median errors for all sketches collapse toward zero on Frequent-100 and Random-100 tokens (Table 3, middle block), and even the infrequent bucket tightens considerably compared with $R = 2^{10}$.
- The visual traces in Figure 2 confirm that widening the sketch curbs most overestimation events for Count-Min and Count-Sketch; Count-Median still exhibits occasional spikes on rare tokens because its unsigned counters cannot cancel collisions.

Observations for $R = 2^{18}$.



(a) Frequent-100 (b) Random-100 (c) Infrequent-100

Figure 3: Relative-error curves for $R = 2^{18}$.

- With the widest sketches, all medians drop to zero and the error curves flatten, indicating the structures now recover the true counts on almost every probe (Table 3, bottom block).
- Residual deviations (Figure 3) stem from the few tokens that still hash-collide; the signed nature of Count-Sketch keeps its spikes smallest whenever they appear.

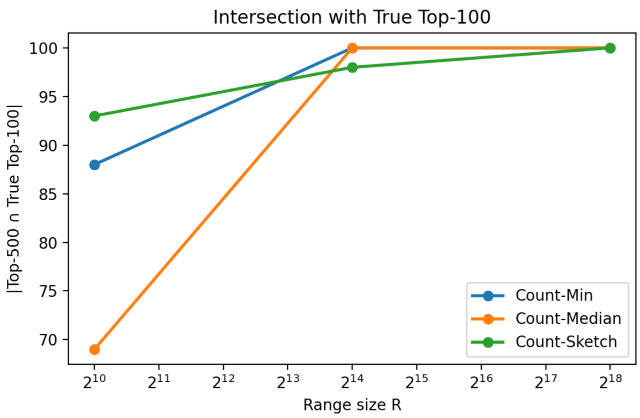


Figure 4: Intersection size of sketch top-500 with true top-100 across R.

Sketch	$R = 2^{10}$	$R = 2^{14}$	$R = 2^{18}$
Frequent-100 median relative error			
Count-Min	0.357	0.007	0
Count-Median	0.578	0.016	0
Count-Sketch	0.072	0.002	0
Random-100 median relative error			
Count-Min	3127.75	46	0
Count-Median	4764	109.5	1.5
Count-Sketch	1	1	0.417
Infrequent-100 median relative error			
Count-Min	4145	79	0
Count-Median	7008.5	196.5	3
Count-Sketch	1	1	1

Table 3: Median relative errors across sketch families and R.

5 Top-500 Intersection

The heap-based tracker yields the set intersection sizes summarised in Table 4. Small values of R drop many true heavy hitters, while widening to $R = 2^{14}$ markedly improves overlap for every sketch in this sample.

Sketch	$R = 2^{10}$	$R = 2^{14}$	$R = 2^{18}$
Count-Min	88	100	100
Count-Median	69	100	100
Count-Sketch	93	98	100

Table 4: Size of $\text{Top-500}_{\text{sketch}} \cap \text{Top-100}_{\text{truth}}$.

6 Reproducibility Checklist

- **Generate outputs**: `python main_a2.py`
- **Artifacts**: Plots land in `outputs/a2/` with filenames `errors_R{R}_{category}.png` and `top500_intersection.png`; metrics appear in `outputs/a2/summary.json`. Each run also writes `error_table.tex`, `median_table.tex`, and `run_summary.tex` so the report stays numerically consistent with the latest metrics—no manual edits required.

7 Conclusions

The combined pipeline is as structured as: it streams the AOL log once, maintains exact frequencies for evaluation, compares three sketches at multiple width settings, quantifies relative errors for representative token buckets, and evaluates top-k recovery. The Count-Min sketch offers deterministic upper bounds but requires larger widths to suppress overestimation on sparse items, Count-Median provides unbiased point estimates at the cost of higher variance (especially with small R), and Count-Sketch trades reduced bias for manageable variance through signed updates.