

CS848 Fall 2026: Indexing for Modern Data Processing

Sketches

Count-Min, AMS, and HyperLogLog

Xiao Hu

September 24, 2026

Announcement

- The presentation schedule is finalized!
- Please review **one** of the papers that will be covered next week:
 - (Approximately) One-page long
 - Summarize the contribution, strong points, weak points, and detailed comments in your viewpoint
 - All paper reviews are submitted on Learn
 - The deadline for submitting the paper review for next week is Sep 28 (Monday) 11:59 EST.

Roadmap

- The streaming setting
 - One pass, sublinear space, approximate answers
- Count-Min sketch: frequency estimation
- AMS sketch: second moment estimation
- HyperLogLog: counting distinct elements
- Where sketches appear in modern systems

- State-of-the-art Research on Concurrent indices -- Tom Iagovet

Recap from last lecture

Guarantees come from the family H , never from a single function h .

What we built

- Universal: $\Pr[h(x) = h(y)] \leq 1/m$.
- k -wise independent: $(h(x_1), h(x_2), \dots, h(x_k))$ uniform on $[m]^k$
 - pairwise independent when $k = 2$
- Universality doesn't imply uniformity, but pairwise independence does.
- k -wise independence implies $(k - 1)$ -wise independence

What it buys today

- Expectation arguments need pairwise-independence.
- Variance arguments need stronger independence.
- Data are adversarial; only our hash choice is random.

The Data Stream Model

Stream	Items arrive one at a time from a universe U .
One pass	Each item is seen once. No rewinding, no random access.
Sublinear space	Memory is strictly much smaller than the stream length and $ U $.
Sketch	A small summary supporting updates and approximate queries.
Mergeable	Sketches of two streams combine into a sketch of the union.

We trade exactness for space, and bound the error we accept.

Notation

Universe

$$U = \{x_1, x_2, \dots, x_{|U|}\}$$

Frequency vector

$\mathbf{a} = (a_1, a_2, \dots, a_{|U|})$, where a_i counts occurrences of item x_i

$$F_0 = \|\mathbf{a}\|_0$$

Number of distinct items, the count of nonzero a_i .

$$F_1 = \|\mathbf{a}\|_1$$

Stream length $n = \sum_{i=1}^{|U|} a_i$ (in the insert-only model).

$$F_2 = \|\mathbf{a}\|_2^2$$

Second moment $\sum_{i=1}^{|U|} a_i^2$. Self-join size, a measure of skew.

ϵ, δ

Error target ϵ and failure probability δ

Three Questions, Three Sketches

How often did x occur?

Count-Min

Point query on a_i

Error $\varepsilon \|a\|_1$

One-sided: never underestimates

How skewed is the stream?

AMS

Second moment $\|a\|_2^2$

Error $\varepsilon \|a\|_2^2$

Two-sided: median of means

How many distinct items?

HyperLogLog

Cardinality $F_0 = \|a\|_0$

Error $\approx 1.04/\sqrt{m}$

Registers of 5 bits each

Same recipe, three different error measures.

1. Build a weak estimator

2. Compute its expectation

3. Apply a concentration bound

4. Replicate for confidence

Why Not Just Count Exactly?

- A hash table of counters answers every query exactly
 - Space is $\Theta(F_0)$ counters. On an IP stream that is billions of keys.
 - A query optimizer needs the answer in microseconds, inside memory it already has.
- Exact distinct counting provably needs $\Omega(F_0)$ bits
 - Any exact algorithm must distinguish all subsets it has seen.
 - So sublinear space forces approximation. This is not laziness.
- **Deal:** fix ε and δ up front, then pay space that depends on ε and δ , not on n and F_0

Preliminary from Probability

- **Markov Inequality:** For any non-negative random variable X and any positive number $a > 0$,

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$$

- **Chebyshev Inequality:** For any random variable X and any positive number $0 < a \leq \sqrt{\text{Var}[X]}$,

$$\Pr[|X - \mathbb{E}[X]| \geq a] \leq \frac{\text{Var}[X]}{a^2}$$

- **Chernoff Bound:** For k independent random variable $x_1, x_2, \dots, x_k$ taking values in $\{0,1\}$, let $X = \sum_{i=1}^k x_i$. For some $0 < \delta$,

$$\Pr[X \geq (1 + \delta)\mathbb{E}[X]] \leq \exp\left(\frac{-\delta^2 \cdot \mathbb{E}[X]}{(2 + \delta)}\right)$$

The Count-Min Sketch

Frequency Estimation

Increment(x_i)

Record one more occurrence of x_i .

Estimate(x_i)

Return $\hat{a}_i$, an approximation of the true count a_i .

Goal

$a_i \leq \hat{a}_i \leq a_i + \varepsilon \|\mathbf{a}\|_1$ with probability at least $1 - \delta$.
(safe elimination; no false negative)

Adversarial data

The frequencies are fixed, not random. Only h is random.

Count-Min: The Structure

A 2D array of d rows, m counters per row,

A pairwise-independent hash family H that maps U to $\{0, 1, \dots, m - 1\}$

Choose a random hash function h from H for each row

h_1	0	0	7	0	0	3	0	0
h_2	0	5	0	0	0	0	9	0
h_3	0	0	0	7	0	0	0	2

m counters

Total space: $d \cdot m$ counters, independent of n and F_0

Highlighted cells are the ones item x_i touches.

Count-Min: Update and Query

- After seeing item x_i
 - for $j = 1 \dots d$:
 - $\text{count}[j][h_j(x_i)] += 1$
- Return the estimator for x_i
 - return $\min(\text{count}[j][h_j(x_i)] \text{ for } j = 1 \dots d)$

Each row gives one estimate of a_i . Every row overcounts, never undercounts, because other items can only add to the cell that x reads.

Update and query both cost $\Theta(d)$ work.

Why Take the Minimum?

- Each row's counter equals a_i plus the counts of items colliding with x_i
 - So every row is an overestimate. The error is one-sided.
- The minimum is therefore the most accurate row
 - It is still an upper bound on a_i , so correctness survives.
 - It fails only if every row is polluted, and rows are independent.
- **Contrast:** with two-sided error the minimum is the worst choice, and we will need the median instead.

Count-Min Analysis: Expected Error

Fix an arbitrary row. Write the counter that $\bar{a}_i$ reads as a_i plus collision noise:

$$\bar{a}_i = a_i + \sum_{k \neq i: k \in [|U|]} a_k \cdot \mathbb{1}[h(x_k) = h(x_i)]$$

Pairwise-independence gives $\Pr[h(x_k) = h(x_i)] \leq 1/m$, so by linearity

$$\mathbb{E}[\bar{a}_i - a_i] = \sum_{k \neq i: k \in [|U|]} a_k \cdot \Pr[h(x_k) = h(x_i)] \leq \frac{\|\mathbf{a}\|_1}{m}$$

The expectation is over the choice of h only.

Count-Min Analysis: Concentration

The error $\bar{a}_i - a_i$ is **nonnegative**, so Markov's inequality applies:

$$\Pr[\bar{a}_i - a_i > \varepsilon \|\mathbf{a}\|_1] \leq \frac{\mathbb{E}[\hat{a}_i - a_i]}{\varepsilon \|\mathbf{a}\|_1} \leq \frac{\frac{\|\mathbf{a}\|_1}{m}}{\varepsilon \|\mathbf{a}\|_1} = \frac{1}{\varepsilon m} = \frac{1}{e}$$

by choosing $m = e/\varepsilon$.

The d rows use independently chosen hash functions.

Let $\bar{a}_{i,j}$ be the estimator returned by the j -th row. Recall that $\hat{a}_i = \min_j \bar{a}_{i,j}$

$$\Pr[\hat{a}_i - a_i > \varepsilon \|\mathbf{a}\|_1] = \Pr[\bar{a}_{i,j} - a_i > \varepsilon \|\mathbf{a}\|_1] \leq \left(\frac{1}{e}\right)^d \leq \delta$$

by choosing $d = \ln(1/\delta)$.

Markov for one-sided error, replication for confidence.

Summary of Count-Min

Parameters

$m = \frac{e}{\varepsilon}$ columns and $d = \ln \frac{1}{\delta}$ rows, each rounded up.

Guarantee

$a_i \leq \hat{a}_i \leq a_i + \varepsilon \|\mathbf{a}\|_1$ with probability $\geq 1 - \delta$.

Space

$\Theta\left(\frac{1}{\varepsilon} \log \frac{1}{\delta} \log m\right)$ bits.

Time

$\Theta\left(\log \frac{1}{\delta}\right)$ per update and per query.

Mergeable

Using the same d, m and H ; sum the corresponding slots.

Weakness

Error scales with $\|\mathbf{a}\|_1$, so rare items drown in a long stream.

Good for heavy hitters, poor for the tail.

Application: Heavy Hitters

- Find every item with $a_i \geq \varphi \|a\|_1$, for example $\varphi = 1\%$
 - Set $\varepsilon = \varphi/2$.
 - Any true heavy hitter has an estimate above $\varphi \|a\|_1$ and will be identified
 - Any item with $a_i < \frac{\varphi}{2} \cdot \|a\|_1$ will not be identified
 - Keep a heap of candidates alongside the sketch, or sweep the universe offline.
- Why the error is acceptable here
 - We only care about items whose counts are a constant fraction of the stream.
 - The $\varepsilon \|a\|_1$ slack is small compared with the threshold $\varphi \|a\|_1$.
- Used for top-k queries, hot key detection, cache admission policy and database query optimization

The *AMS* Sketch

Why the Second Moment?

Definition

$$F_2 = \sum_{i \in [|U|]} a_i^2$$

Self-join size

F_2 is exactly $|R \bowtie R|$ on the join key. Optimizers want it.

Skew measure

F_2/F_1 is large when a few keys dominate.

Join sizing

Cauchy-Schwarz bounds $|R \bowtie S|$ via $\sqrt{F_2(R) \cdot F_2(S)}$

Partition quality

Predicts the worst partition in a hash join or shuffle.

F_2 is the statistic a query planner actually needs.

The Tug-of-War Estimator

Draw s from a **4-wise independent** hashing family with $s: U \rightarrow \{-1, +1\}$.

Keep one counter Z :

- Update: $Z \leftarrow Z + s(x)$ on seeing x
- Estimate: return $Y = Z^2$

Stream: a, b, a, c, a, b. Frequencies 3, 2, 1, so $F_2 = 9 + 4 + 1 = 14$.

One sign draw:

$s(a) = +1$

$s(b) = -1$

$s(c) = +1$

arrival

a

b

a

c

a

b

Z after

+1

0

+1

+2

+3

+2

Estimator $Y = Z^2 = 4$

Averaging Over the Sign Draws

All 8 sign patterns for the same stream, with $Z = 3 \cdot s(a) + 2 \cdot s(b) + 1 \cdot s(c)$:

$s(a)$	$s(b)$	$s(c)$	Z	$Y = Z^2$
+1	+1	+1	6	36
+1	+1	-1	4	16
+1	-1	+1	2	4
+1	-1	-1	0	0
-1	+1	+1	0	0
-1	+1	-1	-2	4
-1	-1	+1	-4	16
-1	-1	-1	-6	36

Unbiased

Mean of Y over the 8 draws:
 $(36+16+4+0+0+4+16+36)/8 = 14$.

High variance

Y ranges from 0 to 36.
Averaging k copies shrinks the spread by a factor of k .

AMS is Unbiased

After the whole stream:

$$Z = \sum_{i \in [|U|]} s(i) \cdot a_i$$

$$Y = Z^2 = \left(\sum_{i \in [|U|]} s(i) \cdot a_i \right)^2 = \sum_{i \in [|U|]} a_i^2 \cdot s^2(i) + \sum_{i,j \in [|U|]: i \neq j} a_i \cdot a_j \cdot s(i) \cdot s(j)$$

$$\mathbb{E}[Y] = \mathbb{E}[Z^2] = \mathbb{E}\left[\sum_{i \in [|U|]} a_i^2 \cdot s^2(i) \right] + \mathbb{E}\left[\sum_{i,j \in [|U|]: i \neq j} a_i \cdot a_j \cdot s(i) \cdot s(j) \right] = F_2$$

$s(i)^2 = 1$ always, so the first sum is exactly F_2

For $i \neq j$, pairwise independence gives $\mathbb{E}[s(i)s(j)] = 0$.

Unbiasedness needs only 2-independence of the sign hash.

AMS: Variance

$$\text{Var}[Y] = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2$$

$$= \mathbb{E} \left[\left(\sum_{i \in [|U|]} s(i) \cdot a_i \right)^4 \right] - F_2^2$$

$$= \mathbb{E} \left[\sum_{i,j,k,l \in [|U|]} s(i)s(j)s(k)s(l)a_i a_j a_k a_l \right] - F_2^2$$

$$= \sum_{i \in [|U|]} a_i^4 + 3 \sum_{i,j \in [|U|]: i \neq j} a_i^2 a_j^2 - F_2^2$$

$$= 2 \sum_{i,j \in [|U|]: i \neq j} a_i^2 a_j^2 \leq 2F_2^2$$


$$\mathbb{E} \left[\sum_{i,j,k,l \in [|U|]: i \neq j \neq k \neq l} s(i)s(j)s(k)s(l) \right] = 0$$

This step asks for 4-wise independence.

AMS: Concentration

Recall Chebyshev inequality: $\Pr[|X - \mathbb{E}[X]| \geq t] \leq \frac{\text{Var}[X]}{t^2}$

Average $k = 6/\varepsilon^2$ independent trials: $\bar{Y} = \frac{1}{k}(Y_1 + Y_2 + \dots + Y_k)$

$$\mathbb{E}[\bar{Y}] = \mathbb{E}\left[\frac{1}{k}(Y_1 + Y_2 + \dots + Y_k)\right] = \frac{1}{k} \sum_{i \in [k]} \mathbb{E}[Y_i] = F_2$$

$$\Pr[|\bar{Y} - F_2| > \varepsilon F_2] \leq \frac{\text{Var}[\bar{Y}]}{(\varepsilon F_2)^2} \leq \frac{2F_2^2}{k\varepsilon^2 F_2^2} = \frac{1}{3}$$

AMS: Concentration

Recall Chernoff bound: For k independent random variable $x_1, x_2, \dots, x_k$ taking values in $\{0,1\}$, let $X = \sum_{i=1}^k x_i$. For some $0 < \delta$,

$$\Pr[X \geq (1 + \delta)\mathbb{E}[X]] \leq \exp\left(\frac{-\delta^2 \cdot \mathbb{E}[X]}{(2 + \delta)}\right)$$

Take the median across $t = 25 \log \frac{1}{\delta}$ independent trials. The median is bad only if half the trials are bad.

For the i -th trial, we introduce a random variable $x_i \in \{0,1\}$ to indicate its failure. Note that $X = \sum_{i=1}^t x_i$ with $\mathbb{E}[X] \leq \frac{t}{3}$. Then,

$$\Pr\left[X \geq \frac{t}{2}\right] \leq \Pr\left[X \geq \frac{3}{2}\mathbb{E}[X]\right] \leq \exp(-\log 1/\delta) = \delta$$

Summary of AMS

Parameters	$k = \Theta(1/\epsilon^2)$ counters per group, $t = \Theta(\log 1/\delta)$ groups.
Guarantee	Estimate lies in $(1 \pm \epsilon)F_2$ with probability $\geq 1 - \delta$.
Space	$\Theta(\epsilon^{-2} \log \delta^{-1})$ counters, plus $\Theta(\log U)$ bits per hash.
Time	$\Theta(\epsilon^{-2} \log \delta^{-1})$ per update and per query.
Mergeable	Using the same sign hashes; sum the counters.
Weakness	Two-sided error; reports F_2 only, not single frequencies.

Unbiased by 2-independence, concentrated by 4-independence.

Count Sketch = Count-Min + Sign Function

Stream	Turnstile updates (x, Δ) , where Δ may be positive or negative .
Goal	Estimate the final frequency a_i of any item x_i .
Challenge	Collisions can now increase or decrease a counter. Count-Min's one-sided minimum no longer applies.
Idea	Give each item a random sign. Signed collision errors cancel in expectation.

Signed hashing supports the general turnstile model.

Count Sketch: Structure

Each row has a bucket hash h and an independent sign hash s .

row 1: $h_1, s_1=+1$

0	0	$+\Delta$	0	0	0	0	0
---	---	-----------	---	---	---	---	---

row 2: $h_2, s_2=-1$

0	0	0	0	0	$-\Delta$	0	0
---	---	---	---	---	-----------	---	---

row 3: $h_3, s_3=+1$

0	$+\Delta$	0	0	0	0	0	0
---	-----------	---	---	---	---	---	---

Update for x_i and $\Delta(x_i)$:

$$C[j, h_j(x_i)] \leftarrow C[j, h_j(x_i)] + s_j(x_i) \Delta(x_i)$$

Query for a_i :

$$\hat{a}_i = \text{median} \{s_j(x_i) \cdot C[j, h_j(x_i)]\}$$

Summary of Count Sketch

Unbiased

Random signs turn collision bias into zero-mean noise.

Can you show the estimator is unbiased?

Parameters

$m = \Theta(\epsilon^{-2})$ buckets and $d = \Theta(\log \delta^{-1})$ independent rows.

Guarantee

$|\hat{a}_i - a_i| \leq \epsilon \|\mathbf{a}\|_2$ with probability at least $1 - \delta$.

Can you show the concentration bound?

Space

$\Theta(\epsilon^{-2} \log \delta^{-1})$ counters, plus seeds for bucket and sign hashes.

Strength

Supports positive and negative updates; useful for ℓ_2 heavy hitters and sparse recovery.

HyperLogLog

Counting Distinct Elements

add(x)	Record that x has been seen.
estimate	Return an approximation of F_0 , the number of distinct items.
Exact cost	$\Omega(F_0)$ bits. A hash set of 10^9 URLs does not fit in cache.
Where it is needed	NDV for the optimizer, unique visitors, distinct IPs, GROUP BY sizing.
Target	Relative error, since F_0 spans many orders of magnitude.

Here we want relative error, not additive error.

From Hashes to Coin Flips

Hash each item and read the bits of $h(x)$ as coin flips. Let $\rho(x)$ be the position of the first 1 bit, so $\Pr[\rho(x) = k] = 2^{-k}$ for a random hash.

0001 0110 1110 0101

three leading zeros, so $\rho = 4$

Seeing a pattern with $\rho = k$ suggests we have seen about 2^k distinct items.

Duplicates hash to the same value, so they replay the same game and change nothing. *(That is exactly the invariance we need for counting distinct items!)*

Rare patterns are evidence of many distinct trials.

A First Estimator, and its Flaws

Track $S = \max p(x)$ over the stream, then report 2^S .

Why it is cheap

Hashes need $\Theta(\log |U|)$ bits.

We store only a bit position, so $\Theta(\log \log |U|)$ bits.

One register covers a universe of 2^{32} in 5 bits.

Why it is bad

The answer is always a power of two.

One lucky long run inflates the estimate.

Variance is high, and the bias is upward.

Cheap and unbiased in spirit, but far too noisy alone.

Stochastic Averaging: One Update

Use the first b bits of $h(x)$ to pick a register ($m = 2^b$), and the rest to play the game.

1. Hash

$$h(x) = 101|0001\dots$$

2. Route

$$j = 101_2 = 5$$

3. Measure

$$\rho(0001\dots) = 4$$

4. Update

$$M[5] \leftarrow \max(M[5], 4)$$

Prefix

The first b bits select exactly one of $m = 2^b$ registers.

Suffix

The remaining bits measure a rare leading-zero event.

Duplicate

The same item repeats the same j and ρ , so the max does not change.

Stochastic Averaging: Reading the Registers

Each register keeps the maximum ρ among the items routed to it.

3	5	2	7	4	3	6	2	5	4
$M[0]$	$M[1]$	$M[2]$	$M[3]$	$M[4]$	$M[5]$	$M[6]$	$M[7]$	$M[8]$	$M[9]$

Registers hold values up to about 32, so 5 bits each. With $m = 2048$ that is 1.5 kB.

Using the same h , sketches of two streams merge register-wise.

$M[3] = 7$ A rare pattern with six leading zeros appeared. This is strong evidence for a larger substream.

$M[2] = 2$ No long run appeared yet. This register provides weaker evidence.

Combine Do not average the $M[j]$ values. HLL combines the reciprocal scales $2^{-M(j)}$.

The HyperLogLog Estimate

$$E = \alpha_m \cdot \frac{m^2}{(2^{-M[1]} + 2^{-M[2]} + \dots + 2^{-M[m]})}$$

- Harmonic mean** Small estimates dominate, so single long runs cannot inflate E.
- α_m** A debiasing constant, about $0.7213/(1 + 1.079/m)$ for $m \geq 128$.
- Accuracy** Relative standard error $\approx \frac{1.04}{\sqrt{m}}$. With $m = 2048$, about 2.3%.
- Corrections** Linear counting for small F_0 ; 64-bit hashes avoid large-range fixes.

1.5 kB buys 2% relative error on any real stream.

HLL Analysis 1: A Single Register

$$\Pr[M \leq k] = (1 - 2^{-k})^n \approx \exp(-n \cdot 2^{-k})$$

Distribution

One item exceeds level k with probability 2^{-k} . All n items stay below k with the probability above.

Threshold

The distribution changes near $n \cdot 2^{-k} \approx 1$, so $M \approx \log_2 n$.

First estimate

Therefore 2^M has the correct scale, but one register has high variance and upward bias.

The maximum reaches the level where the expected number of items surviving to that level becomes approximately one.

HLL Analysis 2: Split Across Registers

$$n_j \approx \frac{F_0}{m} \Rightarrow M[j] \approx \log_2\left(\frac{F_0}{m}\right)$$

Routing

The first b hash bits choose one of $m = 2^b$ registers. Each register sees about F_0/m distinct items.

Local evidence

From $M[j] \approx \log_2(F_0/m)$, we obtain $2^{-M(j)} \approx m/F_0$.

Aggregate

Summing m reciprocal estimates gives $\sum_j 2^{-M(j)} \approx m^2/F_0$.

Invert the aggregate to recover F_0 .

HLL Analysis 3: The Harmonic Mean

$$\text{Estimator} = \alpha_m \cdot \frac{m^2}{\sum_j 2^{-M(j)}}$$

Interpretation

This is α_m times m times the harmonic mean of the register estimates $2^{M(j)}$.

Robustness

One unusually large $M[j]$ contributes almost zero to the denominator, so it cannot dominate the estimate.

Bias correction

α_m corrects systematic bias; for $m \geq 128$, $\alpha_m \approx 0.7213/(1 + 1.079/m)$.

Accuracy

$\text{RSE} \approx \frac{1.04}{\sqrt{m}}$, hence $m \approx (\frac{1.04}{\varepsilon})^2$. With $m = 2048$ registers, the relative standard error is about 2.3%.

HyperLogLog in Practice

Redis	PFADD, PFCOUNT, PFMERGE. 12 kB per sketch at 0.81% error.
Warehouses	APPROX_COUNT_DISTINCT in BigQuery, Snowflake, Presto, Spark.
Optimizers	NDV statistics for join ordering and GROUP BY sizing.
Mergeable	Union of streams = register-wise max. Perfect for shuffles.
Limitation	No intersections. Inclusion-exclusion amplifies the error badly.

Mergeability is why HyperLogLog won in distributed systems.

Takeaways

1

Sublinear space forces approximation, so we fix ϵ and δ and pay for accuracy.

2

Count-Min: one-sided error, min across rows, error $\epsilon \|a\|_1$, space $\epsilon^{-1} \log \delta^{-1}$.

3

AMS: unbiased tug-of-war, 4-independence for variance, median of means.

4

HyperLogLog: max leading zeros per register, harmonic mean, $1.04/\sqrt{m}$ error.

5

All three follow the same four-step recipe, and all are mergeable.

Next Week

- Sep 28 Monday
 - Submit the review to Review Week 4 on Learn
- Sep 29 Tuesday
 - In person at DC 2568
 - Bloom & Cuckoo Filters by Kahbod Amini
- Oct 1 Thursday
 - In person at DC 2568
 - Learned Indexes by Oscar Rodney