20: Counting Distinct Elements

my office hours today at 2pm
in CSE 12203

www.slido.com/2226110
Conditional Expectation

**Definition.** Let \( X \) be a discrete random variable then the **conditional expectation** of \( X \) given event \( A \) is

\[
\mathbb{E}[X \mid A] = \sum_{x \in \Omega_x} x \cdot P(X = x \mid A)
\]

**Note:**
- Linearity of expectation still applies here
  \[
  \mathbb{E}[aX + bY + c \mid A] = a \mathbb{E}[X \mid A] + b \mathbb{E}[Y \mid A] + c
  \]
Law of Total Expectation

**Law of Total Expectation (event version).** Let $X$ be a random variable and let events $A_1, ..., A_n$ partition the sample space. Then,

$$
\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X \mid A_i] \cdot P(A_i)
$$

**Law of Total Expectation (random variable version).** Let $X$ be a random variable and $Y$ be a discrete random variable. Then,

$$
\mathbb{E}[X] = \sum_{y \in \Omega_Y} \mathbb{E}[X \mid Y = y] \cdot P(Y = y)
$$

$A_i$'s $= \{ Y = y_i \}$
Law of total probability for continuous random variables.

**Definition.** Let $A$ be an event and $Y$ a continuous random variable. Then

$$P[A] = \int_{-\infty}^{\infty} P(A|Y = y)f_Y(y)\,dy$$
Data mining – Stream Model

• In many data mining situations, data often not known ahead of time.
  – Examples: Google queries, Twitter or Facebook status updates, YouTube video views
• Think of the data as an infinite stream
• Input elements (e.g. Google queries) enter/arrive one at a time.
  – We cannot possibly store the stream.

Question: How do we make critical calculations about the data stream using a limited amount of memory?
Stream Model – Problem Setup

**Input:** sequence (aka. “stream”) of \( N \) elements \( x_1, x_2, \ldots, x_N \) from a known universe \( U \) (e.g., 8-byte integers).

**Goal:** perform a computation on the input, in a single left to right pass, where:

- Elements processed in real time
- Can’t store the full data \( \Rightarrow \) use minimal amount of storage while maintaining working “summary”
What can we compute?

32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4

Some functions are easy:
- Min
- Max
- Sum
- Average
Today: Counting distinct elements

32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4

Application

You are the content manager at YouTube, and you are trying to figure out the distinct view count for a video. How do we do that?

Note: A person can view their favorite videos several times, but they only count as 1 distinct view!
Other applications

- **IP packet streams**: How many distinct IP addresses or IP flows (source+destination IP, port, protocol)
  - Anomaly detection, traffic monitoring

- **Search**: How many distinct search queries on Google on a certain topic yesterday

- **Web services**: how many distinct users (cookies) searched/browsed a certain term/item
  - Advertising, marketing trends, etc.
Counting distinct elements

\[32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4\]

\(N = \text{# of IDs in the stream} = 11, \quad m = \text{# of distinct IDs in the stream} = 5\)

Want to compute number of \textbf{distinct} IDs in the stream.

- \textbf{Naïve solution:} As the data stream comes in, store all distinct IDs in a hash table.
- \textbf{Space requirement:} \(\Omega(m)\)

\textit{YouTube Scenario: } \(m\) is huge!
Counting distinct elements

32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4

\[ N = \# \text{ of IDs in the stream} = 11, \quad m = \# \text{ of distinct IDs in the stream} = 5 \]

Want to compute number of distinct IDs in the stream.

How to do this without storing all the elements?
Detour – I.I.D. Uniforms

If $Y_1, \ldots, Y_m \sim \text{Unif}(0,1)$ (i.i.d.) where do we expect the points to end up?

$E(Y) = \frac{1}{2}$
Detour – I.I.D. Uniforms

If $Y_1, \ldots, Y_m \sim \text{Unif}(0,1)$ (i.i.d.) where do we expect the points to end up?

$m = 1$

$m = 2$

$\mathbb{E}(\min(Y_1, Y_2)) = \frac{1}{3}$
Detour – I.I.D. Uniforms

If \( Y_1, \ldots, Y_m \sim \text{Unif}(0,1) \) (i.i.d.) where do we expect the points to end up?

“Evenly spread out”

\[
E\left(\min(Y_1, \ldots, Y_4)\right) = \frac{1}{5}
\]
Detour – Min of I.I.D. Uniforms

If $Y_1, \ldots, Y_m \sim \text{Unif}(0,1)$ (iid) where do we expect the points to end up?

In general, $\mathbb{E}[\min(Y_1, \ldots, Y_m)] = \frac{1}{m+1}$

- $\mathbb{E}[\min(Y_1)] = \frac{1}{2} = \frac{1}{2}$
- $\mathbb{E}[\min(Y_1, Y_2)] = \frac{1}{3} = \frac{1}{3}$
- $\mathbb{E}[\min(Y_1, \ldots, Y_4)] = \frac{1}{5} = \frac{1}{5}$
Detour – Min of I.I.D. Uniforms

If \( Y_1, \ldots, Y_m \sim \text{Unif}(0,1) \) (iid) where do we expect the points to end up?

In general, \( \mathbb{E}[\min(Y_1, \ldots, Y_m)] = \frac{1}{m+1} \)

What is some intuition for this?
Detour – Min of I.I.D. Uniforms

If $Y_1, \ldots, Y_m \sim \text{Unif}(0,1)$ (i.i.d.) where do we expect the points to end up?

e.g., what is $\mathbb{E}[\min\{Y_1, \ldots, Y_m\}]$?

CDF: Observe that $\min\{Y_1, \ldots, Y_m\} \geq y$ if and only if $Y_1 \geq y, \ldots, Y_m \geq y$

\[
P(\min\{Y_1, \ldots, Y_m\} \geq y) = P(Y_1 \geq y, \ldots, Y_m \geq y)
= P(Y_1 \geq y) \cdots P(Y_m \geq y) \quad \text{(Independence)}
= (1 - y)^m
\]

$\Rightarrow P(\min\{Y_1, \ldots, Y_m\} \leq y) = 1 - (1 - y)^m$
\[ F_y(y) = \mathbb{P}(Y \leq y) = 1 - (1-y)^m \]

\[ f_y(y) = \frac{d}{dy} F_y(y) = m (1-y)^{m-1} \]

\[ E(Y) = \int_0^y y f_y(y) \, dy = \int_0^y y m (1-y)^{m-1} \, dy \]

\[ = \frac{s}{m+1} \]
Detour – Min of I.I.D. Uniforms

**Useful fact.** For any random variable \( Y \) taking non-negative values

\[
\mathbb{E}[Y] = \int_0^\infty P(Y \geq y) dy
\]

**Proof**

\[
\mathbb{E}[Y] = \int_0^\infty x \cdot f_Y(x) \, dx = \int_0^\infty \left( \int_0^\infty f_Y(x) \, dx \right) \cdot f_Y(x) \, dx = \int_0^\infty \int_0^\infty f_Y(x) \, dy \, dx
\]

\[
= \int_0^\infty \int_y^\infty f_Y(x) \, dx \, dy = \int_0^\infty P(Y \geq y) \, dy
\]
Detour – Min of I.I.D. Uniforms

**Useful fact.** For any random variable $Y$ taking non-negative values

$$E[Y] = \int_0^\infty P(Y \geq y)dy$$

$$E[Y] = \int_0^\infty P(Y \geq y)dy = \int_0^1 (1 - y)^m dy$$

$$= - \frac{1}{m+1} (1 - y)^{m+1}\bigg|_0^1 = 0 - \left(- \frac{1}{m+1}\right) = \frac{1}{m+1}$$

$Y_1, \ldots, Y_m \sim \text{Unif}(0,1)$ (i.i.d.)

$Y = \min\{Y_1, \ldots, Y_m\}$

27
Detour – Min of I.I.D. Uniforms

If $Y_1, \ldots, Y_m \sim \text{Unif}(0,1)$ (iid) where do we expect the points to end up?

In general, $\mathbb{E}[\min(Y_1, \ldots, Y_m)] = \frac{1}{m+1}$

- For $m = 1$,
  $\mathbb{E}[\min(Y_1)] = \frac{1}{1+1} = \frac{1}{2}$

- For $m = 2$,
  $\mathbb{E}[\min(Y_1, Y_2)] = \frac{1}{2+1} = \frac{1}{3}$

- For $m = 4$,
  $\mathbb{E}[\min(Y_1, \ldots, Y_4)] = \frac{1}{4+1} = \frac{1}{5}$
Back to counting distinct elements

32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4

\[ N = \# \text{ of IDs in the stream} = 11, \quad m = \# \text{ of distinct IDs in the stream} = 5 \]

Want to compute number of distinct IDs in the stream.

*How to do this without storing all the elements?*
Distinct Elements – Hashing into $[0, 1]$

Hash function $h : U \rightarrow [0, 1]$

Assumption: For all $x \in U$, $h(x) \sim \text{Unif}(0, 1)$ and mutually independent

$h(32), h(12), h(14), h(32), h(7), h(12), h(32), h(7)$

$0.38, 0.71, 0.15, 0.38, 0.25, 0.71$
Distinct Elements – Hashing into \([0, 1]\)

Hash function \(h: U \rightarrow [0,1]\)

Assumption: For all \(x \in U\), \(h(x) \sim \text{Unif}(0,1)\) and mutually independent

\[
32, \quad 12, \quad 14, \quad 32, \quad 7, \quad 12, \quad 32, \quad 7
\]

\[
h(32), h(12), h(14), h(32), h(7), h(12), h(32), h(7)
\]

4 distinct elements

\[\rightarrow 4 \text{ i.i.d. RVs} \quad h(32), h(12), h(14), h(7) \sim \text{Unif}(0,1)\]

\[\rightarrow \mathbb{E}[\min\{h(32), h(12), h(14), h(7)\}] = \frac{1}{4+1} = \frac{1}{5}\]
Distinct Elements – Hashing into [0, 1]

**Hash function** \( h: U \rightarrow [0,1] \)

**Assumption:** For all \( x \in U \), \( h(x) \sim \text{Unif}(0,1) \) and mutually independent

\[
x_1, x_2, \ldots, x_N \text{ contains } m \text{ distinct elements}
\]

\[
h(x_1), h(x_2), \ldots, h(x_N) \text{ contains } m \text{ i.i.d. rvs } \sim \text{Unif}(0,1)
\]

\[
\mathbb{E}[\min\{ h(x_1), \ldots, h(x_N) \}] = \frac{1}{m + 1}
\]

and \( N - m \) repeats
A super duper clever idea!!!!

$$\mathbb{E}[\min\{h(x_1), \ldots, h(x_N)\}] = \frac{1}{m + 1}$$

So

$$m = \frac{1}{\mathbb{E}[\min\{h(x_1), \ldots, h(x_N)\}]} - 1$$

What if $\min\{h(x_1), \ldots, h(x_N)\}$ is $\approx \mathbb{E}[\min\{h(x_1), \ldots, h(x_N)\}]$?
The MinHash Algorithm – Idea

1. Compute \( \text{val} = \min\{h(x_1), \ldots, h(x_N)\} \)

2. Assume that \( \text{val} \approx \mathbb{E}[\min\{h(x_1), \ldots, h(x_N)\}] \)

3. Output as estimate for \( m \): \( \text{round} \left( \frac{1}{\text{val}} - 1 \right) \)

\[
m = \frac{1}{\mathbb{E}[\min\{h(x_1), \ldots, h(x_N)\}]} - 1
\]
The MinHash Algorithm – Implementation

Algorithm \textit{MinHash}(x_1, x_2, \ldots, x_N)

\begin{align*}
\text{val} & \leftarrow \infty \\
\text{for } i & = 1 \text{ to } N \text{ do} \\
\text{val} & \leftarrow \min\{\text{val}, h(x_i)\} \\
\text{return} & \text{round}\left(\frac{1}{\text{val}} - 1\right)
\end{align*}

Memory cost = just remember \text{val} (with sufficient precision)

\[ \text{val} = \min(h(x_1), \ldots, h(x_N)) \]
MinHash Example

Stream: 13, 25, 19, 25, 19, 19
Hashes: 0.51, 0.26, 0.79, 0.26, 0.79, 0.79

What does MinHash return?

1. Compute $\text{val} = \min\{h(x_1), \ldots, h(x_N)\}$
2. Assume that $\text{val} \approx \mathbb{E}[\min\{h(x_1), \ldots, h(x_N)\}]$
3. Output $\text{round} \left( \frac{1}{\text{val}} - 1 \right)$

Poll: [www.slido.com/2226110](http://www.slido.com/2226110)

- a. 1
- b. 3
- c. 5
- d. No idea
MinHash Example II

Stream: 11, 34, 89, 11, 89, 23
Hashes: 0.5, 0.21, 0.94, 0.5, 0.94, 0.1

Output is $\frac{1}{0.1} - 1 = 9$

Clearly, not a very good answer!

Not unlikely: $P(h(x) < 0.1) = 0.1$
The MinHash Algorithm – Problem

Algorithm \textbf{MinHash}(x_1, x_2, ..., x_N)

val ← ∞

\textbf{for} i = 1 \textbf{to} N \textbf{do}

\quad val ← \min\{val, h(x_i)\}

\textbf{return} \ \text{round}\left(\frac{1}{\text{val}} - 1\right)

But val is not \(\mathbb{E}[\text{val}]\)!

How far is \text{val} from \(\mathbb{E}[\text{val}]\)?

\text{Var}(\text{val}) \approx \frac{1}{(m + 1)^2}

\text{val} = \min\{h(x_1), ..., h(x_N)\}

\mathbb{E}[\text{val}] = \frac{1}{m + 1}
How can we reduce the variance?

**Idea: Repetition to reduce variance!**

Use $k$ independent hash functions $h^1, h^2, \ldots, h^k$

\[
\begin{align*}
\text{val}^1 &= \min(h^1(x), \ldots, h^1(x_N)) = \min(Y^1_1, \ldots, Y^1_m) \\
\text{val}^2 &= \min(h^2(x), \ldots, h^2(x_N)) = \min(Y^2_1, \ldots, Y^2_m) \\
\text{val}^k &= \min(h^k(x), \ldots, h^k(x_N)) = \min(Y^k_1, \ldots, Y^k_m)
\end{align*}
\]

\[
\begin{align*}
E(\text{val}) &= \frac{1}{k} \sum_{i=1}^{k} E(\text{val}^i) = \frac{\frac{1}{m+1}}{k} \\
\text{Var}(\text{val}) &= \frac{1}{k^2} \text{Var}(\frac{1}{k} \sum_{i=1}^{k} \text{val}^i) = \frac{1}{k^2} \frac{\frac{1}{m+1}^2}{(m+1)^2}
\end{align*}
\]
How can we reduce the variance?

Idea: Repetition to reduce variance!
Use $k$ independent hash functions $h^1, h^2, \ldots, h^k$

Algorithm $\text{MinHash}(x_1, x_2, \ldots, x_N)$

\[
\begin{align*}
\text{val}_1, \ldots, \text{val}_k & \leftarrow \infty \\
\text{for } i = 1 \text{ to } N \text{ do} & \\
\quad \text{val}_1 & \leftarrow \min\{\text{val}_1, h^1(x_i)\}, \ldots, \\
\quad \text{val}_k & \leftarrow \min\{\text{val}_k, h^k(x_i)\} \\
\text{val} & \leftarrow \frac{1}{k} \sum_{i=1}^{k} \text{val}_i \\
\text{return } & \text{round}\left(\frac{1}{\text{val}} - 1\right)
\end{align*}
\]

\[
\text{Var}(\text{val}) = \frac{1}{k} \frac{1}{(m + 1)^2}
\]

\[
m = \frac{1}{E(\text{minhash}) - 1}
\]
**MinHash and Estimating # of Distinct Elements in Practice**

- **MinHash in practice:**
  - One also stores the element that has the minimum hash value for each of the $k$ hash functions
  - Then, just given separate MinHashes for sets $A$ and $B$, can also estimate
    - what fraction of $A \cup B$ is in $A \cap B$; i.e., how similar $A$ and $B$ are