Distinct elements

Anna Karlin

With many slides by Luxi Wang, Shreya Jayaraman, Alex Tsun and Jeff Ullman

Data mining

- In many data mining situations, the data is not known ahead of time.

- Examples:
    - Google queries
    - Twitter or Facebook status updates
    - Youtube video views

- In some ways, best to think of the data as an infinite stream that is non-stationary (distribution changes over time)

Stream model

- 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?

Problem

- Input: sequence 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. => use minimal amount of storage while maintaining working “summary”

Counting distinct elements

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

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,

- Want to compute number of **distinct** keys in the stream.
- *How to do this without storing all the elements?*
- *Yet another super cool application of probability (and hashing)*

A naive solution, counting!

Store the n **distinct** user IDs in a hash table.

Space requirement: O(n)

Considering the number of users of YouTube, and the number of videos on YouTube, this is not feasible.

Consider a hash function $h : \mathcal{U} \to [0, 1]$
For distinct values in $\mathcal{U}$, the function maps to iid (independent and identically distributed) Unif(0,1) random numbers.

Note that, if you were to feed in two equivalent elements, the function returns the **same** number.

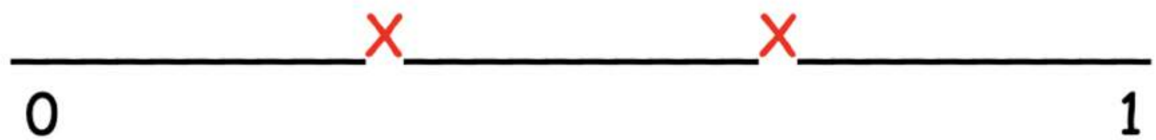
32, 12, 14, 32, 7, 12, 32, 7, 32, 12, 4,

Min of IID Uniforms

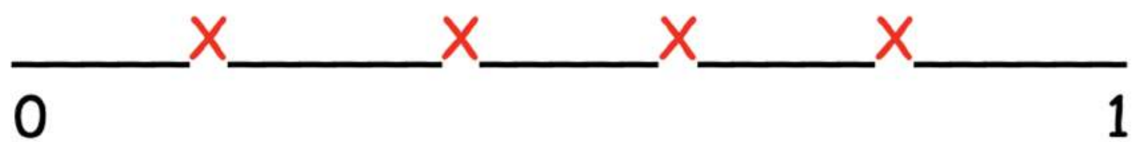
If $Y_1, \ldots, Y_m$ are iid $Unif(0,1)$, where do we "expect" the points to end up?

$m = 1$

$m = 2$



$m = 4$

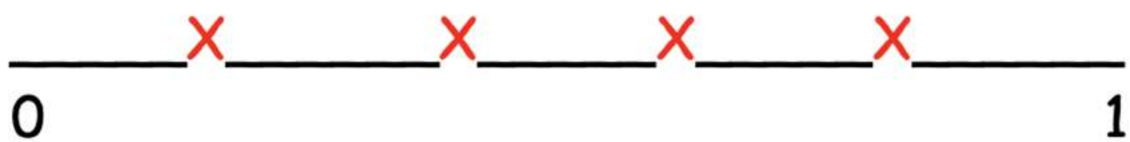


Min of IID Uniforms

If $Y_1, \dots, Y_m$ are iid $Unif(0,1)$, where do we "expect" the points to end up?

$$E[\min\{Y_1, \dots, Y_4\}] = \frac{1}{4+1} = \frac{1}{5}$$

$m = 4$



Min of IID Uniforms

If $Y_1, \ldots, Y_m$ are iid $Unif(0,1)$, where do we "expect" the points to end up?

A super duper clever idea

32 5 17 32 14 5 32 32 17

The Distinct Elements Algorithm

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow \min\{\text{val}, \text{hash}(x)\}$
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$
for $i = 1, \ldots, N$: **do** ▷ Loop through all stream elements
  update(x_i) ▷ Update our single float variable
return estimate() ▷ An estimate for n, the number of distinct elements.

Distinct Elements Example

```
Stream:  13,   25,   19,   25,   19,   19

Hashes: 0.51, 0.26, 0.79, 0.26, 0.79, 0.79
```

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$
for $i = 1, \ldots, N$: **do** ▹ Loop through all stream elements
  update(x_i) ▹ Update our single float variable
return estimate() ▹ An estimate for n, the number of distinct elements.

```
val = infty
```

Distinct Elements Example

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

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$

for $i = 1, \ldots, N$: **do** ▷ Loop through all stream elements
  update(x_i) ▷ Update our single float variable
return estimate() ▷ An estimate for n, the number of distinct elements.

`val = infty`

Distinct Elements Example

Stream: 13, 25, 19, 25, 19, 19

h

Hashes: 0.51, 0.26, 0.79, 0.26, 0.79, 0.79

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$

for $i = 1, \ldots, N$: **do** ▷ Loop through all stream elements
  update(x_i) ▷ Update our single float variable
return estimate() ▷ An estimate for n, the number of distinct elements.

val = 0.51

Distinct Elements Example

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

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$
for $i = 1, \ldots, N$: **do** ▹ Loop through all stream elements
  update(x_i) ▹ Update our single float variable
return estimate() ▹ An estimate for n, the number of distinct elements.

val = 0.26

Distinct Elements Example

Stream: 13, 25, 19, 25, 19, 19

h

Hashes: 0.51, 0.26, 0.79, 0.26, 0.79, 0.79

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$

function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}

function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$

for $i = 1, \ldots, N$: **do** ▹ Loop through all stream elements
  update(x_i) ▹ Update our single float variable
return estimate() ▹ An estimate for n, the number of distinct elements.

val = 0.26

Distinct Elements Example

Stream: 13, 25, 19, 25, 19, 19

h

Hashes: 0.51, 0.26, 0.79, 0.26, 0.79, 0.79

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$
for $i = 1, \ldots, N$: **do** ▹ Loop through all stream elements
  update(x_i) ▹ Update our single float variable
return estimate() ▹ An estimate for n, the number of distinct elements.

val = 0.26

Distinct Elements Example

Stream: 13, 25, 19, 25, 19, 19

h

Hashes: 0.51, 0.26, 0.79, 0.26, 0.79, 0.79

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$
for $i = 1, \ldots, N$: **do** ▷ Loop through all stream elements
  update(x_i) ▷ Update our single float variable
return estimate() ▷ An estimate for n, the number of distinct elements.

val = 0.26

Distinct Elements Example

Stream: 13, 25, 19, 25, 19, $\boxed{19}$

h

Hashes: 0.51, 0.26, 0.79, 0.26, 0.79, $\boxed{0.79}$

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
 val $\leftarrow \infty$
function UPDATE(x)
 val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
 return round $\left(\frac{1}{\text{val}} - 1\right)$
for $i = 1, \ldots, N$: **do** ▷ Loop through all stream elements
 update(x_i) ▷ Update our single float variable
return estimate() ▷ An estimate for n, the number of distinct elements.

val = 0.26

Distinct Elements Example

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

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$
for $i = 1, \ldots, N$: **do** ▷ Loop through all stream elements
  update(x_i) ▷ Update our single float variable
return estimate() ▷ An estimate for n, the number of distinct elements.

val = 0.26

Return
round(1/0.26 - 1) =
round(2.846) =
3

DIY: Distinct Elements Example II

```
Stream: 11,   34,   89,   11,   89,   23

Hashes: 0.5, 0.21, 0.94, 0.5, 0.94, 0.1
```

Algorithm 2 Distinct Elements Operations

function INITIALIZE()
  val $\leftarrow \infty$
function UPDATE(x)
  val $\leftarrow$ min {val, hash(x)}
function ESTIMATE()
  return round $\left(\frac{1}{\text{val}} - 1\right)$

for $i = 1, \ldots, N$: **do** ▷ Loop through all stream elements
  update(x_i) ▷ Update our single float variable
return estimate() ▷ An estimate for n, the number of distinct elements.

val = 0.1

Return= 9

Summary so far

Problem

How can we reduce the variance?

Coding on pset 6

You will use a hash function $h : \mathcal{U} \rightarrow [0, 1]$
For distinct values in $\mathcal{U}$, the function maps to iid (independent and identically distributed) Unif(0,1) random numbers.

Note that, if you were to feed in two equivalent elements, the function returns the **same** number.

We will implement the hash function for you! Just know that you can consider it an iid uniform continuous random variables for each of the values being hashed.

To do better...

1. we will keep track of K DistElts classes each with its own independent hash function
2. take the mean of our K mins to get a better estimate of the min
3. and then apply the same trick as earlier to give an estimate for the number of distinct elements based on this min that we saw.

Joint Distributions

Anna Karlin
Most slides by Alex Tsun

5.1 Joint Discrete Distributions

Agenda

- Motivation
- Cartesian Products of Sets
- Joint PMFs and Expectation
- Marginal PMFs

Naive Bayes Classifier - What we Calculate

$$\mathbb{P}(\text{spam} \mid \text{"You buy Viagra!"}) = \frac{\mathbb{P}(\text{"You buy Viagra!"} \mid \text{spam})\ \mathbb{P}\ (\text{spam})}{\mathbb{P}(\text{"You buy Viagra!"})}$$

$$= \frac{\mathbb{P}(\{\text{"you"},\text{"buy"},\text{"viagra"}\} \mid \text{spam})\ \mathbb{P}(\text{spam})}{\mathbb{P}(\{\text{"you"},\text{"buy"},\text{"viagra"}\} \mid \text{spam})\ \mathbb{P}(\text{spam}) + \mathbb{P}(\{\text{"you"},\text{"buy"},\text{"viagra"}\} \mid \text{ham})\ \mathbb{P}(\text{ham})} \quad [\text{LTP}]$$

Naive Bayes Classifier - The naive part

$$\mathbb{P}(\{\text{"you"}, \text{"buy"}, \text{"viagra"}\} \mid \text{spam})$$
$$\approx \mathbb{P}(\text{"you"} \mid \text{spam})\mathbb{P}(\text{"buy"} \mid \text{spam})\mathbb{P}(\text{"viagra"} \mid \text{spam})$$

Why is this Naive?

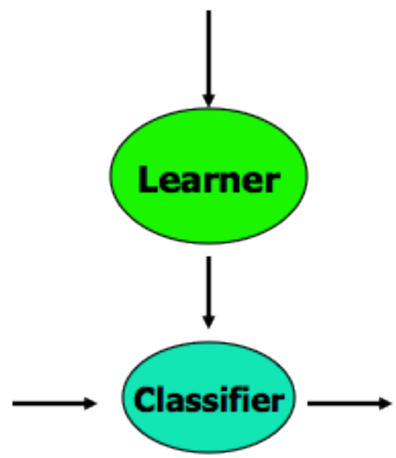
`“!!!Lunch free for You. Viagra included.  $$$ !!!!!”`

Ubiquitous in ML

- Given "labeled data"

Temp.	BP.	Sore Throat	...	Colour	diseaseX
35	95	Y	...	Pale	No
22	110	N	...	Clear	Yes
:	:			:	:
10	87	N	...	Pale	No

- Learn CLASSIFIER, that can predict label of *NEW* instance



Temp	BP	Sore-Throat	...	Color	diseaseX
32	90	N	...	Pale	?

diseaseX
No

Agenda

- Cartesian Products of Sets
- Joint PMFs and Expectation
- Marginal PMFs

Cartesian Product of Sets

Cartesian Product: Let A, B be sets. The Cartesian product of A and B is denoted

$$A \times B = \{(a, b): a \in A, b \in B\}$$

A small example:

$$\{1,2,3\} \times \{4,5\} = \{(1,4), (1,5), (2,4), (2,5), (3,4), (3,5)\}$$

Another example: The xy-plane (2D space) is denoted

$$\mathbb{R}^2 = \mathbb{R} \times \mathbb{R} = \{(x, y): x \in \mathbb{R}, y \in \mathbb{R})\}$$

If A, B are finite sets, then $|A \times B| = |A| \cdot |B|$ by the product rule of counting.

Example: weird dice Again

Suppose I roll two fair 4-sided die independently. Let X be the value of the blue die, and Y the value of the red die. Specify

$\Omega_X = \{1,2,3,4\}$ $\Omega_Y = \{1,2,3,4\}$

$$\Omega_{X,Y} = \Omega_X \times \Omega_Y$$

Specify the joint PMF $p_{X,Y}(x,y) = P(X = x, Y = y)$ for $x, y \in \Omega_{X,Y}$.

X\Y	1	2	3	4
1				
2				
3				
4				

Example: weird dice Again

Suppose I roll two fair 4-sided die independently. Let X be the value of the blue die, and Y the value of the red die. Specify

$$\Omega_X = \{1,2,3,4\} \qquad \Omega_Y = \{1,2,3,4\}$$

$$\Omega_{X,Y} = \Omega_X \times \Omega_Y$$

Specify the joint PMF $p_{X,Y}(x,y) = P(X = x, Y = y)$ for $x, y \in \Omega_{X,Y}$.

$$p_{X,Y}(x,y) = \begin{cases} 1/16, & x,y \in \Omega_{X,Y} \\ 0, & \text{otherwise} \end{cases}$$

X\Y	1	2	3	4
1	1/16	1/16	1/16	1/16
2	1/16	1/16	1/16	1/16
3	1/16	1/16	1/16	1/16
4	1/16	1/16	1/16	1/16

Joint PMFs and Expectation

Joint PMFs: Let X, Y be discrete random variables. The joint PMF of X and Y is

$$p_{X,Y}(a,b) = P(X = a, Y = b)$$

The joint range is

$$\Omega_{X,Y} = \{(c,d): p_{X,Y}(c,d) > 0\} \subseteq \Omega_X \times \Omega_Y$$

Note that

$$\sum_{(s,t)\in\Omega_{X,Y}} p_{X,Y}(s,t) = 1$$

Example: weird dice Again

Suppose I roll two fair 4-sided die independently. Let X be the value of the blue die, and Y the value of the red die. Let $U = \min\{X, Y\}$ and $V = \max\{X, Y\}$.

$\Omega_U = \{1,2,3,4\}$ $\Omega_V = \{1,2,3,4\}$

$\Omega_{U,V} = \{(u, v) \in \Omega_U \times \Omega_V : u \leq v\} \neq \Omega_U \times \Omega_V$

U\V	1	2	3	4
1				
2				
3				
4				

Example: weird dice Again

Suppose I roll two fair 4-sided die independently. Let X be the value of the blue die, and Y the value of the red die. Let $U = \min\{X,Y\}$ and $V = \max\{X,Y\}$.

$\Omega_U = \{1,2,3,4\}$ $\Omega_V = \{1,2,3,4\}$

$\Omega_{U,V} = \{(u,v) \in \Omega_U \times \Omega_V \colon u \leq v\} \neq \Omega_U \times \Omega_V$

Specify the joint PMF $p_{U,V}(u,v) = P(U=u, V=v)$ for $u,v \in \Omega_{U,V}$.

U\V	1	2	3	4
1				
2				
3				
4				

Example: weird dice Again

Suppose I roll two fair 4-sided die independently. Let X be the value of the blue die, and Y the value of the red die. Let $U = \min\{X, Y\}$ and $V = \max\{X, Y\}$.

$\Omega_U = \{1,2,3,4\}$ $\qquad\qquad$ $\Omega_V = \{1,2,3,4\}$

$\Omega_{U,V} = \{(u, v) \in \Omega_U \times \Omega_V : u \le v\} \neq \Omega_U \times \Omega_V$

Specify the joint PMF $p_{U,V}(u, v) = P(U = u, V = v)$ for $u, v \in \Omega_{U,V}$.

$$p_{U,V}(u,v) = \begin{cases} 2/16, & u, v \in \Omega_U \times \Omega_V, \quad v > u \\ 1/16, & u, v \in \Omega_U \times \Omega_V, \quad v = u \\ 0, & \text{otherwise} \end{cases}$$

U\V	1	2	3	4
1	1/16	2/16	2/16	2/16
2	0	1/16	2/16	2/16
3	0	0	1/16	2/16
4	0	0	0	1/16

Example: weird dice Again

Suppose I roll two fair 4-sided die independently. Let X be the value of the blue die, and Y the value of the red die. Let $U = \min\{X, Y\}$ and $V = \max\{X, Y\}$.

What is $p_U(u)$ for $u \in \Omega_U$?

$$p_U(u) = \begin{cases} & u = 1 \\ & u = 2 \\ & u = 3 \\ & u = 4 \end{cases}$$

U\V	1	2	3	4
1	1/16	2/16	2/16	2/16
2	0	1/16	2/16	2/16
3	0	0	1/16	2/16
4	0	0	0	1/16

Example: weird dice Again

Suppose I roll two fair 4-sided die independently. Let X be the value of the blue die, and Y the value of the red die. Let $U = \min\{X, Y\}$ and $V = \max\{X, Y\}$.

What is $p_U(u)$ for $u \in \Omega_U$?

$$p_U(u) = \begin{cases} 7/16, & u = 1 \\ 5/16, & u = 2 \\ 3/16, & u = 3 \\ 1/16, & u = 4 \end{cases}$$

U\V	1	2	3	4
1	1/16	2/16	2/16	2/16
2	0	1/16	2/16	2/16
3	0	0	1/16	2/16
4	0	0	0	1/16

Example: weird dice Again

Suppose I roll two fair 4-sided die independently. Let X be the value of the blue die, and Y the value of the red die. Let $U = \min\{X, Y\}$ and $V = \max\{X, Y\}$.

U\V	1	2	3	4
1	1/16	2/16	2/16	2/16
2	0	1/16	2/16	2/16
3	0	0	1/16	2/16
4	0	0	0	1/16

Joint PMFs and Expectation

Joint PMFs: Let X, Y be discrete random variables. The joint PMF of X and Y is

$$p_{X,Y}(a,b) = P(X = a, Y = b)$$

The joint range is

$$\Omega_{X,Y} = \{(c,d) : p_{X,Y}(c,d) > 0\} \subseteq \Omega_X \times \Omega_Y$$

Note that

$$\sum_{(s,t)\in\Omega_{X,Y}} p_{X,Y}(s,t) = 1$$

If $g: \mathbb{R}^2 \to \mathbb{R}$ is a function, then

$$E[g(X,Y)] = \sum_x \sum_y g(x,y) p_{X,Y}(x,y)$$

.

Marginal PMFs

Marginal PMFs: Let X, Y be discrete random variables. The marginal PMF of X is

$$p_X(a) = \sum_{b \in \Omega_Y} p_{X,Y}(a, b)$$

Marginal PMFs

Marginal PMFs: Let X, Y be discrete random variables. The marginal PMF of X is

$$p_X(a) = \sum_{b \in \Omega_Y} p_{X,Y}(a, b)$$

Similarly, the marginal PMF of Y is

$$p_Y(d) = \sum_{c \in \Omega_X} p_{X,Y}(c, d)$$

Marginal PMFs

Marginal PMFs: Let X, Y be discrete random variables. The marginal PMF of X is

$$p_X(a) = \sum_{b \in \Omega_Y} p_{X,Y}(a, b)$$

Similarly, the marginal PMF of Y is

$$p_Y(d) = \sum_{c \in \Omega_X} p_{X,Y}(c, d)$$

(Extension) If Z is also a discrete random variable, then the marginal PMF of Z is

$$p_Z(z) = \sum_{x \in \Omega_X} \sum_{y \in \Omega_Y} p_{X,Y,Z}(x, y, z)$$

Independence

<u>**Independence (DRVs):**</u> Discrete random variables X, Y are independent, written $X \perp Y$, if for all $x \in \Omega_X$ and $y \in \Omega_Y$,

$$p_{X,Y}(x, y) = p_X(x) p_Y(y)$$

Recall $\Omega_{X,Y} = \{(x, y): p_{X,Y}(x, y) > 0\} \subseteq \Omega_X \times \Omega_Y$. A necessary but not sufficient condition for independence is that $\Omega_{X,Y} = \Omega_X \times \Omega_Y$. That is, if $\Omega_{X,Y} \neq \Omega_X \times \Omega_Y$, then X and Y cannot be independent, but if $\Omega_{X,Y} = \Omega_X \times \Omega_Y$, then we have to check the condition.

This is because if there is some $(a, b) \in \Omega_X \times \Omega_Y$ but not in $\Omega_{X,Y}$, then $p_{X,Y}(a, b) = 0$ but $p_X(a) > 0$ and $p_Y(b) > 0$, violating independence.

Variance adds for Independent RVs

If X, Y are independent random variables $X \perp Y$, then

$$Var(X+Y) = Var(X) + Var(Y)$$

This property relies on the fact that they are independent, whereas linearity of expectation **always** holds, regardless. If $a, b, c \in \mathbb{R}$ are scalars, then

$$Var(aX + bY + c) = a^2 Var(X) + b^2 Var(Y)$$

If $X \perp Y$, then $E[XY] = E[X]E[Y]$.