Estimation of Entropy in Constant Space

Maryam Aliakbarpour

Boston University and Northeastern University

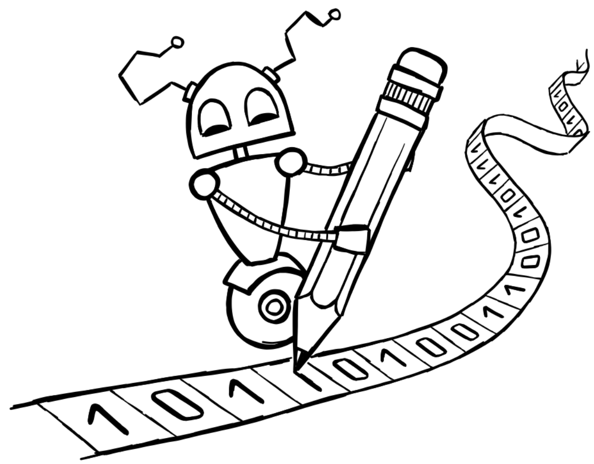
Joint work Andrew McGregor, Jelani Nelson, and Erik Waingarten.

General inference model

Samples from Data distribution → **Algorithm** Test / Estimate / learn → Result

Typical question: How many samples to achieve certain error?

- Unlimited memory and time



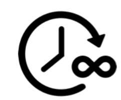


Image from: https://tilics.dmi.unibas.ch/the-turing-machine

General inference model

Samples from
Data distribution → **Algorithm**
Test / Estimate / learn → Result

In application: How many samples if we have limited computational resources?

~~Unlimited memory and time~~

Memory Constraint

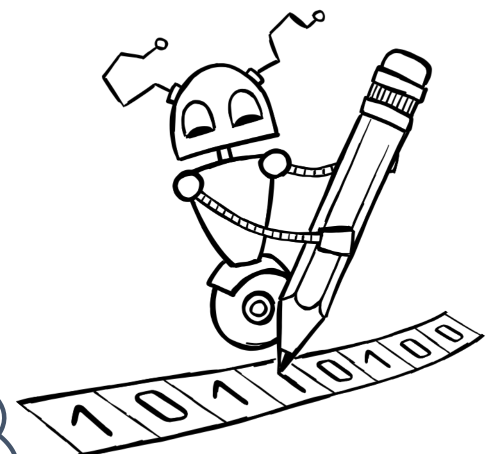


Image from: https://tilics.dmi.unibas.ch/the-turing-machine

Example

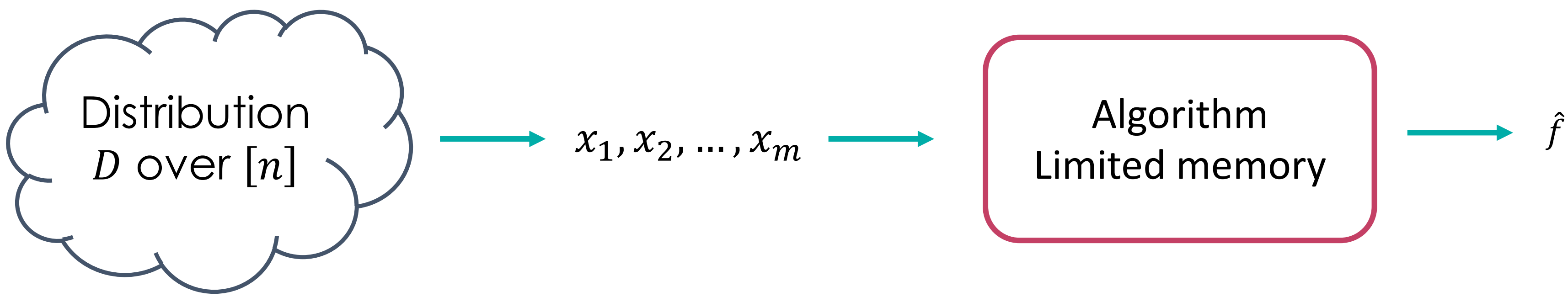
Memory << size of data

Estimation with memory constraints

Unknown distribution D over $[n]$

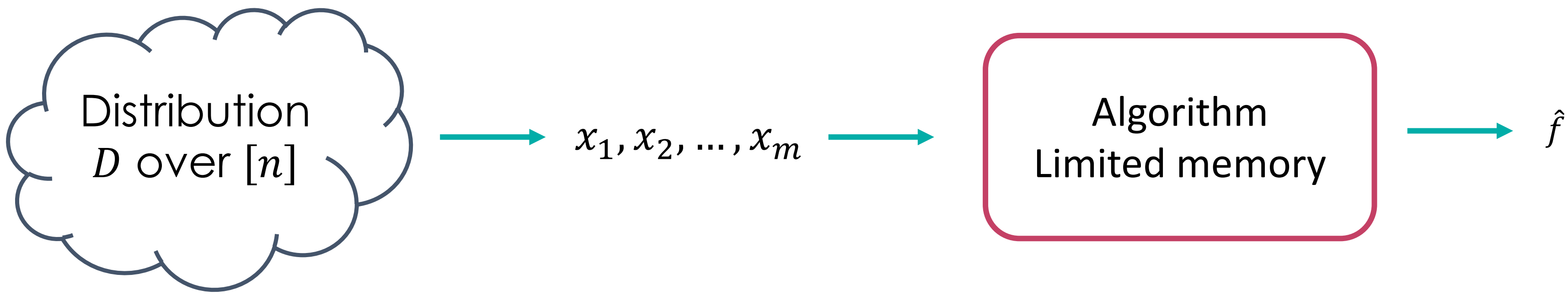
Goal: Estimate $f(D)$ with error ϵ with probability $1 - \delta$ via samples

- (e.g., mean, variance, etc.)



$$\Pr\left[\left|\hat{f} - f(D)\right| > \epsilon\right] \leq \delta$$

A closely related model



This talk: Properties of the distribution

Properties of the data stream

Prior work:

Problems: parity learning, learning PDFs, learning concept classes, robust estimation of statistics, distribution testing, estimating moments,

[Raz. FOCS16][Crouch, McGregor, Valiant, Woodruff, ESA 2016] [Guha, McGregor. AISTATS 2007], [Chien, Ligett, McGregor. ITS 2010] [Steinhardt, Valiant, Wagner. COLT 2016] [Esfandiari, Hajiaghayi, Liaghat, Monemizadeh. SODA 2015] [Moshkovitz, Moshkovitz. COLT 2017] [Kol, Raz, Tal. STOC 2017] [Raz. FOCS 2017] [Garg, Kothari, Raz. STOC 2018] [Sharam, Sidford, Valiant. STOC 2019] [Diakonikolas, Gouleakis, Kane, Rao. COLT 2019] [Garg, Raz, Tal. Complexity 2019] [Acharya, Bhadane, Indyk, Sun, NeurIPS 2019] [Garg, Kothari, Raz. RANDOM 2020] [Garg et al. RANDOM 2021] [Brown, Bun, Smith. COLT 2022]…

This work: estimating entropy

Shannon's Entropy of $D = (p_1, p_2, \dots, p_n)$:

$$H(D) := \sum_{i=1}^{n} p_i \log_2 1/p_i$$

Entropy

Information theory ⋮

In information theory, the entropy of a random variable is the average level of "information", "surprise", or "uncertainty" inherent to the variable's possible outcomes. Wikipedia

Feedback

This work: estimating entropy

Shannon's Entropy of $D = (p_1, p_2, \ldots, p_n)$:

$$H(D) \coloneqq \sum_{i=1}^{n} p_i \log_2 1/p_i$$

$$\begin{array}{c}\text{Samples from } D\\ x_1, x_2, \ldots, x_m\end{array} \longrightarrow \boxed{\text{Algorithm}} \longrightarrow \widehat{H}$$

Goal:

$$\Pr\left[\left|\widehat{H} - H(D)\right| > \epsilon\ \right] \leq 0.1$$

Memory constrains: $O(1)$ words

Previous results

No memory constraint:

$$\Theta\left(\frac{n}{\epsilon \log n} + \frac{\log^2 n}{\epsilon^2}\right)$$

[Batu, Dasgupta, Kumar, Rubinfeld. STOC 2002][Valiant, Valiant. FOCS 2011] [Valiant, Valiant. JACM 2017] [Wu, Yang. IEEE Trans. IT 2016] [Jiao et al. IEEE Trans. IT 2015] …. (and many more)

Previous results

No memory constraint:

$$\Theta\left(\frac{n}{\epsilon \log n} + \frac{\log^2 n}{\epsilon^2}\right)$$

[Batu, Dasgupta, Kumar, Rubinfeld. STOC 2002][Valiant, Valiant. FOCS 2011] [Valiant, Valiant. JACM 2017] [Wu, Yang. IEEE Trans. IT 2016] [Jiao et al. IEEE Trans. IT 2015] …. (and many more)

$O(1)$ words of memory:

$$\mathrm{O}\left(\frac{n \log (1/\epsilon)^3}{\epsilon^3}\right)$$

[Acharya, Bhadane, Indyk, Sun, NeurIPS 2019]

Our results

This work $O(1)$ words of memory:

$$\mathrm{O}\left(\frac{n\log(1/\epsilon)^4}{\epsilon^2}\right)$$

$O(1)$ words of memory:

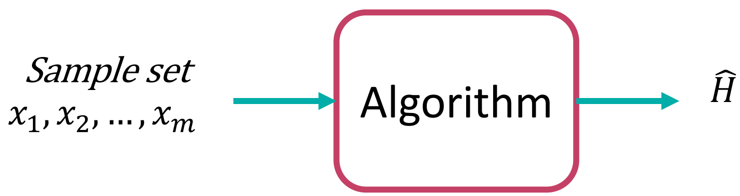
$$\mathrm{O}\left(\frac{n\log{(1/\epsilon)^3}}{\epsilon^3}\right)$$

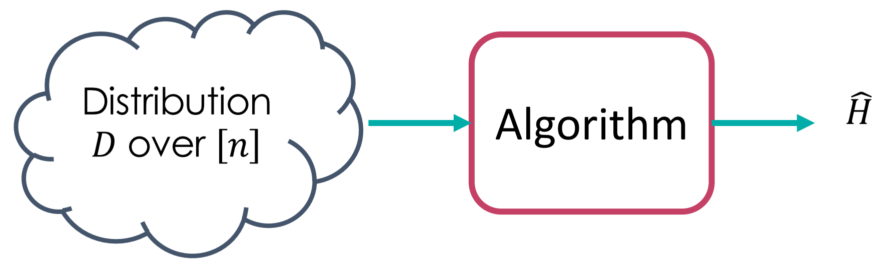
[Acharya, Bhadane, Indyk, Sun, NeurIPS 2019]

A closely related problem

Estimating empirical entropy in the data streaming

vs.





$$\Omega\left(\frac{1}{\epsilon^2}\cdot(\log\log n+\ \log 1/\epsilon)\right) \text{ bits}$$

[Chakrabarti, Cormode, McGregor'10]
[Jayaram Woodruff'19]

Possible $O(1)$ words of memory
= $Polylog(k, 1/\epsilon)$ bits

Entropy estimation with no memory constraint

No memory constraint

Algorithm [Valiant and Valiant'11]:

1. Compute the fingerprint of the samples

   Count numbers of elements appeared i times

List 3 1 3 8 7 3 1 5

Fingerprints

three elements appeared once.

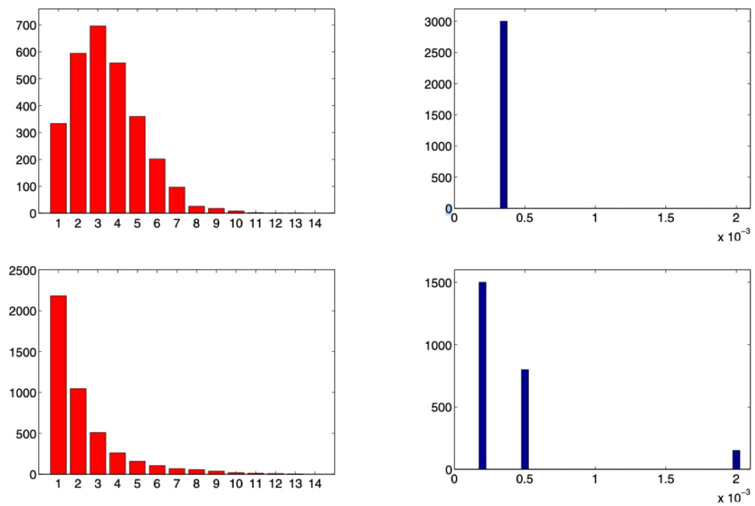
One element appeared twice.

One element appeared three times.

No memory constraint

Algorithm [Valiant, Valiant'11]:

1. Compute the fingerprint of the samples
2. Come up with a histogram of a distribution that is likely to generate



Plots from [Valiant, Valiant'11]

No memory constraint

Algorithm [Valiant, Valiant’11]:

1. Compute the fingerprint of the samples
2. Come up with a histogram of a distribution that is likely to generate
3. Output a distribution that is compatible with the histogram

Works well ignoring the labels!

Entropy ✓

Support size ✓

Adding memory constraints

Computing fingerprint is hard when we cannot memorize

List

3 1 3 8 7 3 1 5

Fingerprints

three elements appeared once.

One element appeared twice.

One element appeared three times.

Entropy estimation with ~~no~~ memory constraint

A simple approach

Simple algorithm

$$H(D) \coloneqq \sum_{i=1}^{n} p_i \cdot \log 1/p_i = \mathrm{E}_{i\sim D}[\log 1/p_i]$$

1. Repeat r times
   1. Draw $i \sim D$.
   2. $\hat{p}_i \leftarrow$ Estimate p_i
2. Output average of $\log 1/\hat{p}_i$'s.

Simple algorithm

$$H(D) := \sum_{i=1}^{n} p_i \cdot \log 1/p_i = \mathrm{E}_{i\sim D}[\log 1/p_i]$$

1. Repeat r times
   1. Draw $i \sim D$.
   2. $\hat{p}_i \leftarrow$ Estimate p_i

   Via negative binomial distribution

   Draw samples until t copies of i are observed.

   $X_i \leftarrow \frac{1}{t} \cdot (\text{\# Observed samples})$

   $\mathrm{E}[X_i]$ is precisely $1/p_i$.
2. Output average of $\log 1/\hat{p}_i$'s.

Simple algorithm

$$H(D) \coloneqq \sum_{i=1}^{n} p_i \cdot \log 1/p_i = \mathrm{E}_{i\sim D}[\log 1/p_i]$$

1. Repeat r times
   1. Draw $i \sim D$.
   2. $\hat{p}_i \leftarrow$ Estimate p_i

   Via negative binomial distribution

   Draw samples until t copies of i are observed.

   $X_i \leftarrow \frac{1}{t} \cdot (\#\ \text{Observed samples})$

   $\mathrm{E}[X_i]$ is precisely $1/p_i$.
2. Output average of $\log 1/\hat{p}_i$'s.

Simple algorithm: Analysis of error

$$H(D) - \widehat{H} = H(D) - \mathrm{E}\big[\widehat{H}\big] + \mathrm{E}\big[\widehat{H}\big] - \widehat{H}$$

output

$$= \underbrace{\mathrm{E}_{i\sim D}[\log 1/p_i] - \mathrm{E}_{i\sim D}[\log X_i]}_{\text{Bias}} + \underbrace{\mathrm{E}\big[\widehat{H}\big] - \widehat{H}}_{\text{Error of estimation}}$$

$t = \Theta(1/\epsilon)$ implies bias $< \epsilon/2$

$\mathrm{Var}_{\mathrm{i}\sim D}[\log \mathrm{X\_i}] = O(\log^2 n)$

$r = \Theta(\log^2 n/\epsilon^2)$ implies that error $< \epsilon/2$

$$E[\#\text{samples}] = \Theta(r \cdot t \cdot n) = \Theta(n \log^2 n/\epsilon^3)$$

Simple algorithm: Analysis of error

$H(D) - \widehat{H}$

output

Next:

Making $t = O(\text{polylog}(1/\epsilon))$

Removing extra $\log n$ factors

estimation

$t = \Theta(1/\epsilon)$ implies bias $< \epsilon/2$

$\mathrm{Var}_{i \sim D}[\log X\_i] = O(\log^2 n)$

$r = \Theta(\log^2 n/\epsilon^2)$ implies that error $< \epsilon/2$

$$E[\#\text{samples}] = \Theta(r \cdot t \cdot n) = \Theta(n \log^2 n/\epsilon^3)$$

Entropy estimation with
~~no~~ memory constraint

A ~~simple~~ better approach

Remove bias

Let $Y_i \leftarrow p_i X_i$

Idea: Estimate bias and decrease it from $\widehat{H}$.

$$\text{Bias} = |\mathrm{E}_{i\sim D}[\log 1/p_i] - \mathrm{E}_{i\sim D}[\log X_i]| = |\mathrm{E}_{i\sim D}[\log Y_i]|$$

$\mathrm{E}_{i\sim D}[Y_i]$ = 1. Taylor expansion around Y = 1:

$$\text{Bias} = \mathrm{E}_{i\sim D}[\log Y_i] = \mathrm{E}\left[Y_i - 1 - \frac{(Y_i-1)^2}{2} + \frac{(Y_i-1)^3}{3} - \cdots\right]$$

Remove bias

Idea: Truncated Taylor expansion. Keep the first $s = \log(1/\epsilon)$ terms.

Bias $< \mathrm{E}\Big[$ … $_i - 1)^{s+1}\Big]$

Making $t = O(\mathrm{polylog}(1/\epsilon))$. ✔

Nicely concentrated

Polynomial of degree s of p_i

$\Pr[\text{k samples are equal}] = p_i^k$

Remove $\log n$ factors

Idea: Bu

Partitio

Removing $O(\log n)$.

$\in I_\ell]$

q_ℓ $\quad$ H_ℓ

Estimate $\hat{q}_L$ and $\widehat{H}_L$

Largest bucket $\hat{q}_L = 1 - \sum_{i=1}^{L-1} \hat{q}_\ell$

Error $\leq \left|\sum_{\ell=1}^{L-1}(\hat{q}_\ell - q_\ell)\cdot(H_\ell - H_L)\right| + \left|\sum_{\ell=1}^{L} q_\ell \cdot \left(H_\ell - \widehat{H}_\ell\right)\right|$

Buckets of large X_i can be computed with less accuracy.

Conclusion

This work $O(1)$ words of memory:

$$\mathrm{O}\left(\frac{n\log(1/\epsilon)^4}{\epsilon^2}\right)$$

Open question: can we improve the lower bound to $\Omega\left(\frac{n}{\epsilon^2}\right)$?

Thank you.