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

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

Memory $\ll$ size of data
Estimation with memory constraints

Unknown distribution $D$ over $[n]$

Goal: Estimate $f(D)$ with error $\epsilon$ with probability $1 - \delta$ via samples
- (e.g., mean, variance, etc.)

$$\Pr[|\hat{f} - f(D)| > \epsilon] \leq \delta$$
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,
This work: estimating entropy

Shannon’s Entropy of $D = (p_1, p_2, ..., p_n)$:

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

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](https://en.wikipedia.org/wiki/Entropy)
This work: estimating entropy

Shannon’s Entropy of $D = (p_1, p_2, ..., p_n)$:

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

Goal:

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

Memory constrains: $O(1)$ words
Previous results

No memory constraint:

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

Previous results

No memory constraint:

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


$O(1)$ words of memory:

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

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

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

$$0 \left( \frac{n \log(1/\epsilon)^4}{\epsilon^2} \right)$$

$O(1)$ words of memory:

$$0 \left( \frac{n \log (1/\epsilon)^3}{\epsilon^3} \right)$$

[Acharya, Bhadane, Indyk, Sun, NeurIPS 2019]
A closely related problem

Estimating empirical entropy in the data streaming

\[ \Omega \left( \frac{1}{\varepsilon^2} \cdot (\log \log n + \log 1/\varepsilon) \right) \text{ bits} \]

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

Possible \( O(1) \) words of memory
\[ = \text{Polylog}(k, 1/\varepsilon) \text{ bits} \]
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

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

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

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

2. Output average of \( \log 1/\hat{p}_i \)'s.
Simple algorithm

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

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.
      
      \[ X_i \leftarrow \frac{1}{t} \cdot (\# \text{ Observed samples}) \]
      E\([X_i]\) is precisely \( 1/p_i \).

2. Output average of \( \log 1/\hat{p}_i \)’s.
Simple algorithm

\[ H(D) := \sum_{i=1}^{n} p_i \cdot \log \frac{1}{p_i} = E_{i \sim D}[\log 1/p_i] \]

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.
      
      \( X_i \leftarrow \frac{1}{t} \cdot \) (# Observed samples)
      
      \( E[X_i] \) is precisely \( 1/p_i \).

2. Output average of \( \log 1/\hat{p}_i \)'s.
Simple algorithm: Analysis of error

$$H(D) - \hat{H} = H(D) - E[\hat{H}] + E[\hat{H}] - \hat{H}$$

\[= E_{i \sim D}[\log 1/p_i] - E_{i \sim D}[\log X_i] + E[\hat{H}] - \hat{H}\]

Bias

Error of estimation

output

\[t = \Theta(1/\epsilon) \text{ implies } \text{bias} < \epsilon/2\]

\[\text{Var}_{i \sim D}[\log X_i] = O(\log^2 n)\]

\[r = \Theta(\log^2 n / \epsilon^2) \text{ implies that } \text{error} < \epsilon/2\]

\[E[\#\text{samples}] = \Theta(r \cdot t \cdot n) = \Theta(n \log^2 n / \epsilon^3)\]
Simple algorithm: Analysis of error

\[ H(D) - \hat{H} \]

output

Next:
Making \( t = O(\text{polylog}(1/\epsilon)) \)
Removing extra \( \log n \) factors

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

\( \text{Var}_{i \sim D}[\log X_i] = O(\log^2 n) \)

\( r = \Theta(\log^2 n / \epsilon^2) \) 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

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

Let $Y_i \leftarrow p_i X_i$

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

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

$$\text{Bias} = E_{i \sim D}[\log Y_i] = E \left[ Y_i - 1 - \frac{(Y_i-1)^2}{2} + \frac{(Y_i-1)^3}{3} - \ldots \right]$$
Remove bias

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

\[
\text{Bias} < \mathbb{E} \left[ \sum_i \frac{1}{(i - 1)^{s+1}} \right]
\]

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

Polynomial of degree \( s \) of \( p_i \)

\[
\mathbb{P}[	ext{k samples are equal}] = p_i^k
\]
Remove $\log n$ factors

Idea: Bucketing

Partition the range of $\% \& \log %$ into $'intervals$

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

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

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

Buckets of large $X_i$ can be computed with less accuracy.
Conclusion

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

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

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

Thank you.