# Topics in TCS

## $\ell_0$-sampling

Raphaël Clifford

# Introduction to $\ell_0$ sampling

Over a large data set that assigns counts to tokens, the goal of an $\ell_0$-sampler is to draw (approximately) uniformly from the set of tokens with non-zero frequency.

# Introduction to $\ell_0$ sampling

Over a large data set that assigns counts to tokens, the goal of an $\ell_0$-sampler is to draw (approximately) uniformly from the set of tokens with non-zero frequency.

This is non-trivial because we want to use small space and counts can be both positive and negative.

# Introduction to $\ell_0$ sampling

Over a large data set that assigns counts to tokens, the goal of an $\ell_0$-sampler is to draw (approximately) uniformly from the set of tokens with non-zero frequency.

This is non-trivial because we want to use small space and counts can be both positive and negative.

Consider a stream of visits by customers to the busy website of some business or organization. An analyst might want to sample uniformly from the set of all distinct customers who visited the website. ($\ell_0$-sampling)

# Introduction to $\ell_0$ sampling

Over a large data set that assigns counts to tokens, the goal of an $\ell_0$-sampler is to draw (approximately) uniformly from the set of tokens with non-zero frequency.

This is non-trivial because we want to use small space and counts can be both positive and negative.

Consider a stream of visits by customers to the busy website of some business or organization. An analyst might want to sample uniformly from the set of all distinct customers who visited the website. ($\ell_0$-sampling)

Or an analyst might want to sample with probability proportional to their visit frequency. ($\ell_1$-sampling)

# Approximate $\ell_0$ sampling

The $\ell_0$-sampling cannot be solved exactly in sublinear space deterministically.

# Approximate $\ell_0$ sampling

The $\ell_0$-sampling cannot be solved exactly in sublinear space deterministically.

We will see a randomised approximate algorithm.

# Approximate $\ell_0$ sampling

The $\ell_0$-sampling cannot be solved exactly in sublinear space deterministically.

We will see a randomised approximate algorithm.

Let $\|\boldsymbol{f}\|_0$ be the number of tokens with non-zero frequency. Define the probability for token $i$ as

$$\pi_i = \frac{1}{\|\boldsymbol{f}\|_0}, \text{ if } i \in \operatorname{supp} \boldsymbol{f}$$
$$\pi_i = 0, \text{ otherwise}$$

We assume that $\boldsymbol{f} \neq \mathbf{0}$.

# The overall idea

We will sample substreams randomly in such a way that there is a good chance that one is strictly 1-sparse. We will run a sparse recovery algorithm on each substream.

# The overall idea

We will sample substreams randomly in such a way that there is a good chance that one is strictly 1-sparse. We will run a sparse recovery algorithm on each substream.

Our method for achieving this is called “geometric sampling" as each substream samples tokens with geometrically decreasing probability.

# The overall idea

We will sample substreams randomly in such a way that there is a good chance that one is strictly 1-sparse. We will run a sparse recovery algorithm on each substream.

Our method for achieving this is called "geometric sampling" as each substream samples tokens with geometrically decreasing probability.

We will use our sparse recovery and detection algorithm to report the index of the token with non-zero frequency.

# The overall idea

We will sample substreams randomly in such a way that there is a good chance that one is strictly 1-sparse. We will run a sparse recovery algorithm on each substream.

Our method for achieving this is called “geometric sampling" as each substream samples tokens with geometrically decreasing probability.

We will use our sparse recovery and detection algorithm to report the index of the token with non-zero frequency.

The reported token will be uniformly sampled from all tokens with non-zero frequency.

# $\ell_0$-sampling algorithm

Where $\log n$ is written it should be read as $\lceil \log_2 n \rceil$. We will write $\mathscr{D}_\ell$ for the $\ell$th instance of a 1-sparse recovery algorithm.

```
initialise
for each ℓ from 0 to log n
     choose h_ℓ : [n] → {0,1}^ℓ uniformly at random
     set D_ℓ = 0

process(j, c)
for each ℓ from 0 to log n
     if h_ℓ(j) = 0 then                      # probability 2^(−ℓ)
          feed (j, c) to D_ℓ                 # 1-sparse recovery

output
for each ℓ from 0 to log n
     if D_ℓ reports strictly 1-sparse
          output (i, ℓ) and stop             # token, frequency
output FAIL
```

# $\ell_0$-sampling algorithm example

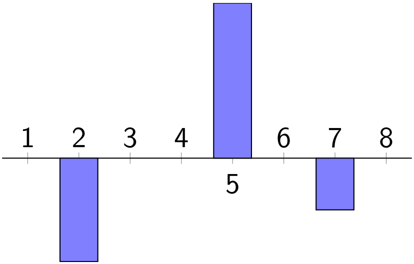


Figure: Frequency vector $\boldsymbol{f}$

- The non-zero frequency item tokens are $2, 5, 7$.

# $\ell_0$-sampling algorithm example

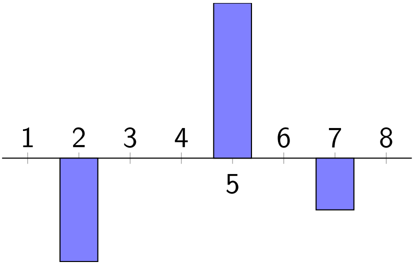


Figure: Frequency vector $\boldsymbol{f}$

| $\ell$ | Prob. | Tokens included |
|---|---|---|
| $\ell = 0$ | 1 | $2, 5, 7$ |
| $\ell = 1$ | $1/2$ | $2, 5$ |
| $\ell = 2$ | $1/4$ | 7 |
| $\ell = 3$ | $1/8$ | 2 |

- The non-zero frequency item tokens are $2, 5, 7$.
- We make 4 substreams.

```
process(j, c)
for each ℓ from 0 to log n
      if h_ℓ(j) = 0 then
            feed (j, c) to D_ℓ
```

# $\ell_0$-sampling algorithm example

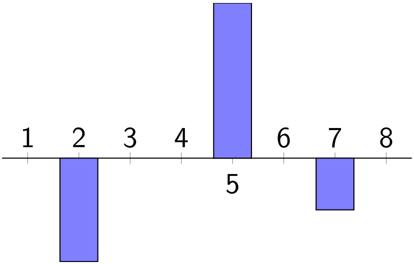


Figure: Frequency vector $\boldsymbol{f}$

| $\ell$ | Prob. | Tokens included |
|---|---|---|
| $\ell = 0$ | 1 | $2, 5, 7$ |
| $\ell = 1$ | $1/2$ | $2, 5$ |
| $\ell = 2$ | $1/4$ | 7 |
| $\ell = 3$ | $1/8$ | 2 |

- The non-zero frequency item tokens are $2, 5, 7$.
- We make 4 substreams.
- With high probability we return 7.

```
process(j, c)
for each ℓ from 0 to log n
      if h_ℓ(j) = 0 then
            feed (j, c) to D_ℓ
```

# $\ell_0$-sampling analysis I

- Let $d = |\text{supp}(\boldsymbol{f})|$. We want to compute a lower bound for the probability that a substream is strictly 1-sparse.

# $\ell_0$-sampling analysis I

- Let $d = |\text{supp}(\boldsymbol{f})|$. We want to compute a lower bound for the probability that a substream is strictly 1-sparse.
- For a fixed level $\ell$, define indicator r.v. $X_j = 1$ if token $j$ is selected in level $\ell$. Let $S = X_1 + \cdots + X_d$. The event that the substream is strictly 1-sparse is $\{S = 1\}$.

# $\ell_0$-sampling analysis I

- Let $d = |\mathrm{supp}(\boldsymbol{f})|$. We want to compute a lower bound for the probability that a substream is strictly 1-sparse.

- For a fixed level $\ell$, define indicator r.v. $X_j = 1$ if token $j$ is selected in level $\ell$. Let $S = X_1 + \cdots + X_d$. The event that the substream is strictly 1-sparse is $\{S = 1\}$.

- We have $\mathbb{E}X_j = p, q = 1 - p$ and $\mathbb{E}(X_j X_k) = p^2$ if $j \neq k$ and $p = p^2 + pq$ otherwise.

# $\ell_0$-sampling analysis I

- Let $d = |\text{supp}(\boldsymbol{f})|$. We want to compute a lower bound for the probability that a substream is strictly 1-sparse.

- For a fixed level $\ell$, define indicator r.v. $X_j = 1$ if token $j$ is selected in level $\ell$. Let $S = X_1 + \cdots + X_d$. The event that the substream is strictly 1-sparse is $\{S = 1\}$.

- We have $\mathbb{E}X_j = p, q = 1 - p$ and $\mathbb{E}(X_j X_k) = p^2$ if $j \neq k$ and $p = p^2 + pq$ otherwise.

- By Chebyshev,

$$\begin{aligned}
\Pr(S \neq 1) = \Pr(|S-1| \geq 1) &\leq \mathbb{E}(S-1)^2 \\
&= \mathbb{E}(S^2) - 2\mathbb{E}(S) + 1 \\
&= \sum_{j,k \in [d]} \mathbb{E}(X_j X_k) - 2 \sum_{j \in [d]} \mathbb{E}(X_j) + 1 \\
&= d^2 p^2 + dpq - 2dp + 1
\end{aligned}$$

# $\ell_0$-sampling analysis II

- $\Pr(S \neq 1) = \Pr(|S - 1| \geq 1) \leq d^2 p^2 + dpq - 2dp + 1.$

# $\ell_0$-sampling analysis II

- $\Pr(S \neq 1) = \Pr(|S - 1| \geq 1) \leq d^2p^2 + dpq - 2dp + 1$.

- The probability that a substream is strictly 1-sparse is therefore at least $2dp - d^2p^2 - dpq = dp(1 - (d-1)p) > dp(1 - dp)$.

# $\ell_0$-sampling analysis II

- $\Pr(S \neq 1) = \Pr(|S-1| \geq 1) \leq d^2p^2 + dpq - 2dp + 1.$

- The probability that a substream is strictly 1-sparse is therefore at least $2dp - d^2p^2 - dpq = dp(1-(d-1)p) > dp(1-dp).$

- If $p = c/d$ for $c \in (0,1)$ then the probability that a substream is strictly 1-sparse is at least $c(1-c)$.

# $\ell_0$-sampling analysis II

- $\Pr(S \neq 1) = \Pr(|S-1| \geq 1) \leq d^2p^2 + dpq - 2dp + 1$.

- The probability that a substream is strictly 1-sparse is therefore at least $2dp - d^2p^2 - dpq = dp(1-(d-1)p) > dp(1-dp)$.

- If $p = c/d$ for $c \in (0,1)$ then the probability that a substream is strictly 1-sparse is at least $c(1-c)$.

- Consider level $\ell$ such that $\frac{1}{4d} \leq \frac{1}{2^\ell} < \frac{1}{2d}$. This constrains $\ell$ to be a unique value for any $d \geq 1$.

# $\ell_0$-sampling analysis II

- $\Pr(S \neq 1) = \Pr(|S-1| \geq 1) \leq d^2p^2 + dpq - 2dp + 1$.
- The probability that a substream is strictly 1-sparse is therefore at least $2dp - d^2p^2 - dpq = dp(1-(d-1)p) > dp(1-dp)$.
- If $p = c/d$ for $c \in (0,1)$ then the probability that a substream is strictly 1-sparse is at least $c(1-c)$.
- Consider level $\ell$ such that $\frac{1}{4d} \leq \frac{1}{2^\ell} < \frac{1}{2d}$. This constrains $\ell$ to be a unique value for any $d \geq 1$.
- We therefore have that the probability that a substream at such a level $\ell$ is strictly 1-sparse is at least $\frac{1}{4}(1-\frac{1}{4}) = 3/16 > 1/8$.

# $\ell_0$-sampling analysis III

- By repeating the whole procedure $O(\log(1/\delta))$ times we reduce the probability that no substream is 1-sparse to $O(\delta)$. To see this, $(\frac{7}{8})^x = \delta \implies x = \log_2(1/\delta)/\log_2(8/7)$.

# $\ell_0$-sampling analysis III

- By repeating the whole procedure $O(\log(1/\delta))$ times we reduce the probability that no substream is 1-sparse to $O(\delta)$. To see this, $(\frac{7}{8})^x = \delta \implies x = \log_2(1/\delta)/\log_2(8/7)$.

- Each run of the 1-sparse algorithm fails with probability $O(1/n^2)$ and so the overall probability of failure is $O(\frac{\log n \log(1/\delta)}{n^2})$.

# $\ell_0$-sampling summary

The $\ell_0$ sampling problem asks us to sample independently and uniformly from the tokens with non-zero frequency.

# $\ell_0$-sampling summary

The $\ell_0$ sampling problem asks us to sample independently and uniformly from the tokens with non-zero frequency.

We use geometric sampling and the 1-sparse recovery and detection algorithm.

# $\ell_0$-sampling summary

The $\ell_0$ sampling problem asks us to sample independently and uniformly from the tokens with non-zero frequency.

We use geometric sampling and the 1-sparse recovery and detection algorithm.

The space is $O(\log n) \cdot O(\log(1/\delta)) \cdot O(\log n + \log M) = O(\log n \cdot \log(1/\delta)(\log n + \log M))$ bits.

# $\ell_0$-sampling summary

The $\ell_0$ sampling problem asks us to sample independently and uniformly from the tokens with non-zero frequency.

We use geometric sampling and the 1-sparse recovery and detection algorithm.

The space is $O(\log n) \cdot O(\log(1/\delta)) \cdot O(\log n + \log M) = O(\log n \cdot \log(1/\delta)(\log n + \log M))$ bits.

The time per arriving token, count pair is $O(\log n \cdot \log(1/\delta))$.

# $\ell_0$-sampling summary

The $\ell_0$ sampling problem asks us to sample independently and uniformly from the tokens with non-zero frequency.

We use geometric sampling and the 1-sparse recovery and detection algorithm.

The space is $O(\log n) \cdot O(\log(1/\delta)) \cdot O(\log n + \log M) = O(\log n \cdot \log(1/\delta)(\log n + \log M))$ bits.

The time per arriving token, count pair is $O(\log n \cdot \log(1/\delta))$.

The probably of failure, because one of the 1-sparse algorithm instances gives a false positive is $O(\frac{\log n \cdot \log(1/\delta)}{n^2})$.

# $\ell_0$-sampling summary

The $\ell_0$ sampling problem asks us to sample independently and uniformly from the tokens with non-zero frequency.

We use geometric sampling and the 1-sparse recovery and detection algorithm.

The space is $O(\log n) \cdot O(\log(1/\delta)) \cdot O(\log n + \log M) = O(\log n \cdot \log(1/\delta)(\log n + \log M))$ bits.

The time per arriving token, count pair is $O(\log n \cdot \log(1/\delta))$.

The probably of failure, because one of the 1-sparse algorithm instances gives a false positive is $O(\frac{\log n \cdot \log(1/\delta)}{n^2})$.

This $\ell_0$-sampling problem will have applications to graph streaming which you will see next.