Topics in TCS

Appoximate counting

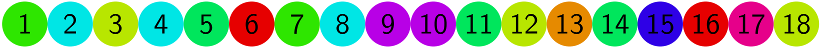
Raphaël Clifford

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?

Approximate counting - Morris

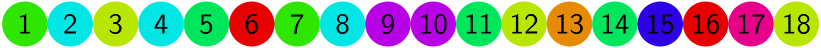
We are going to consider a much simpler problem. How many circles?



Just keep a counter! This needs $O(\log m)$ bits which is optimal.

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?



- Just keep a counter! This needs $O(\log m)$ bits which is optimal.
- So what can we do? Approximate!

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Just keep a counter! This needs $O(\log m)$ bits which is optimal.

So what can we do? Approximate!

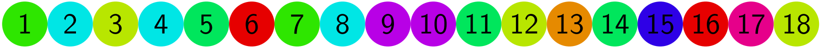
stream $\langle a_1, a_2, \ldots, a_m \rangle$
Set $x = 0$

Morris(a_i)
with probability 2^{-x}
  set x = x + 1

return 2^x-1

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?



Just keep a counter! This needs $O(\log m)$ bits which is optimal.

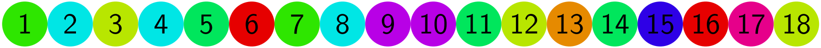
So what can we do? Approximate!

```
stream ⟨a_1, a_2, ..., a_m⟩
Set x = 0

Morris(a_i)
with probability 2^{-x}
   set x = x + 1

return 2^x-1
```

Let's try it on a stream. We get:

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?



Just keep a counter! This needs $O(\log m)$ bits which is optimal.

So what can we do? Approximate!

```
stream ⟨a_1, a_2, ..., a_m⟩
Set x = 0

Morris(a_i)
with probability 2^-x
   set x = x + 1

return 2^x-1
```

Let's try it on a stream. We get:

$x = 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, \ldots$

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?

1 3 3 3 3 7 7 7 7 7 7 7 7 15 15 15 15 15

Just keep a counter! This needs $O(\log m)$ bits which is optimal.

So what can we do? Approximate!

```
stream ⟨a_1, a_2, ..., a_m⟩
Set x = 0

Morris(a_i)
with probability 2^{-x}
   set x = x + 1

return 2^x-1
```

Let's try it on a stream. We get:

$x = 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, \ldots$

These return:
$1, 3, 3, 3, 3, 7, 7, 7, 7, 7, 7, 7, 7, 15, 15, 15, \ldots$

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?

1 3 3 3 3 7 7 7 7 7 7 7 7 15 15 15 15 15

Just keep a counter! This needs $O(\log m)$ bits which is optimal.

So what can we do? Approximate!

stream $\langle a_1, a_2, \ldots, a_m \rangle$
Set $x = 0$

Morris(a_i)
with probability 2^{-x}
  set x = x + 1

return 2^x-1

Let's try it on a stream. We get:

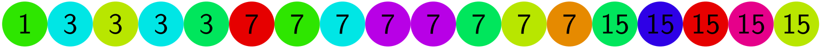
$x = 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, \ldots$

These return:
$1, 3, 3, 3, 3, 7, 7, 7, 7, 7, 7, 7, 7, 15, 15, 15, \ldots$

Running time $O(m)$

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?



Just keep a counter! This needs $O(\log m)$ bits which is optimal.

So what can we do? Approximate!

stream $\langle a_1, a_2, \ldots, a_m \rangle$
Set $x = 0$

Morris(a_i)
with probability 2^{-x}
  set x = x + 1

return 2^x-1

Let's try it on a stream. We get:

$x = 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, \ldots$

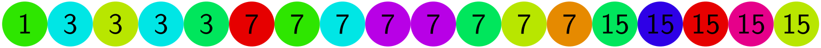
These return:
$1, 3, 3, 3, 3, 7, 7, 7, 7, 7, 7, 7, 7, 15, 15, 15, \ldots$

Running time $O(m)$

Space is ??? (we will see later).

Approximate counting - Morris

We are going to consider a much simpler problem. How many circles?



Just keep a counter! This needs $O(\log m)$ bits which is optimal.

So what can we do? Approximate!

```
stream ⟨a_1, a_2, ..., a_m⟩
Set x = 0

Morris(a_i)
with probability 2^-x
   set x = x + 1

return 2^x-1
```

Let's try it on a stream. We get:

$x = 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, \ldots$

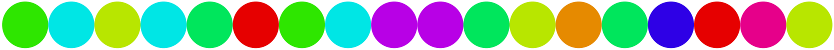
These return:
$1, 3, 3, 3, 3, 7, 7, 7, 7, 7, 7, 7, 7, 15, 15, 15, \ldots$

Running time $O(m)$

Space is ??? (we will see later).

But how accurate is this going to be?

MORRIS - Quality of estimate



Let r.v. $C_n = 2^x$ after n symbols have been read in. We will prove that $\mathbb{E}(C_n) = n + 1$.

MORRIS - Quality of estimate

Let r.v. $C_n = 2^x$ after n symbols have been read in. We will prove that $\mathbb{E}(C_n) = n + 1$.

Consider an equivalent although less space efficient algorithm.

```
stream ⟨a_1, a_2, ..., a_m⟩
Set c = 1

SIMPLIFIED-MORRIS(a_i)
with probability 1/c
   set c = 2c

return c − 1
```

Lemma

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

$\mathrm{var}(C_n) = n(n-1)/2$

MORRIS - Quality of estimate

Let r.v. $C_n = 2^x$ after n symbols have been read in. We will prove that $\mathbb{E}(C_n) = n + 1$.

Consider an equivalent although less space efficient algorithm.

```
stream ⟨a_1, a_2, ..., a_m⟩
Set c = 1

SIMPLIFIED-MORRIS(a_i)
with probability 1/c
   set c = 2c

return c − 1
```

Lemma

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

$\mathrm{var}(C_n) = n(n-1)/2$

MORRIS is therefore an *unbiased* estimator for the number of symbols.

MORRIS - Quality of estimate

Let r.v. $C_n = 2^x$ after n symbols have been read in. We will prove that $\mathbb{E}(C_n) = n + 1$.

Consider an equivalent although less space efficient algorithm.

```
stream ⟨a_1, a_2, ..., a_m⟩
Set c = 1

SIMPLIFIED-MORRIS(a_i)
with probability 1/c
   set c = 2c

return c − 1
```

Lemma

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

$\mathrm{var}(C_n) = n(n-1)/2$

MORRIS is therefore an *unbiased* estimator for the number of symbols.

But we want to know the probability of the estimate being really wrong.

MORRIS - Quality of estimate

Let r.v. $C_n = 2^x$ after n symbols have been read in. We will prove that $\mathbb{E}(C_n) = n + 1$.

Consider an equivalent although less space efficient algorithm.

```
stream ⟨a_1, a_2, ..., a_m⟩
Set c = 1

SIMPLIFIED-MORRIS(a_i)
with probability 1/c
   set c = 2c

return c − 1
```

Lemma

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

$\mathrm{var}(C_n) = n(n-1)/2$

MORRIS is therefore an *unbiased* estimator for the number of symbols.

But we want to know the probability of the estimate being really wrong.

We will need the variance for this.

Morris - Analysis of Expectation

Let us look at the first two symbols that arrive.

Morris - Analysis of Expectation

Let us look at the first two symbols that arrive.

- After the first token, $\mathbb{E}(C_1) = 2$. The variable c will then be set to 2.

MORRIS - Analysis of Expectation

Let us look at the first two symbols that arrive.

- After the first token, $\mathbb{E}(C_1) = 2$. The variable c will then be set to 2.
- After the second token, $\mathbb{E}(C_2) = \frac{1}{2} \cdot 2^2 + \frac{1}{2}2 = 3$. The variable c will be set to 4 with probability $\frac{1}{2}$.

Morris - Analysis of Expectation

Let us look at the first two symbols that arrive.

- After the first token, $\mathbb{E}(C_1) = 2$. The variable c will then be set to 2.
- After the second token, $\mathbb{E}(C_2) = \frac{1}{2} \cdot 2^2 + \frac{1}{2}2 = 3$. The variable c will be set to 4 with probability $\frac{1}{2}$.

Lemma (Expectation of Morris's algorithm)

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

Proof.

Morris - Analysis of Expectation

Let us look at the first two symbols that arrive.

- After the first token, $\mathbb{E}(C_1) = 2$. The variable c will then be set to 2.
- After the second token, $\mathbb{E}(C_2) = \frac{1}{2} \cdot 2^2 + \frac{1}{2}2 = 3$. The variable c will be set to 4 with probability $\frac{1}{2}$.

Lemma (Expectation of Morris's algorithm)

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

Proof.

Let r.v. $Z_i = 1$ if c is increased when the ith symbol arrives and 0 otherwise.

MORRIS - Analysis of Expectation

Let us look at the first two symbols that arrive.

- After the first token, $\mathbb{E}(C_1) = 2$. The variable c will then be set to 2.
- After the second token, $\mathbb{E}(C_2) = \frac{1}{2} \cdot 2^2 + \frac{1}{2}2 = 3$. The variable c will be set to 4 with probability $\frac{1}{2}$.

LEMMA (Expectation of Morris's algorithm)

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

Proof.

Let r.v. $Z_i = 1$ if c is increased when the ith symbol arrives and 0 otherwise.

$\Pr(Z_i = 1) = 1/C_i$ and $C_{i+1} = (1 + Z_i)C_i$.

MORRIS - Analysis of Expectation

Let us look at the first two symbols that arrive.

- After the first token, $\mathbb{E}(C_1) = 2$. The variable c will then be set to 2.
- After the second token, $\mathbb{E}(C_2) = \frac{1}{2} \cdot 2^2 + \frac{1}{2}2 = 3$. The variable c will be set to 4 with probability $\frac{1}{2}$.

LEMMA (Expectation of Morris's algorithm)

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

Proof.

Let r.v. $Z_i = 1$ if c is increased when the ith symbol arrives and 0 otherwise.

$\Pr(Z_i = 1) = 1/C_i$ and $C_{i+1} = (1 + Z_i)C_i$.

If we fix C_i then $\mathbb{E}(1 + Z_i)C_i = (1 + \frac{1}{C_i})C_i = C_i + 1$.

MORRIS - Analysis of Expectation

Let us look at the first two symbols that arrive.

- After the first token, $\mathbb{E}(C_1) = 2$. The variable c will then be set to 2.
- After the second token, $\mathbb{E}(C_2) = \frac{1}{2} \cdot 2^2 + \frac{1}{2}2 = 3$. The variable c will be set to 4 with probability $\frac{1}{2}$.

LEMMA (Expectation of Morris's algorithm)

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

Proof.

Let r.v. $Z_i = 1$ if c is increased when the ith symbol arrives and 0 otherwise.

$\Pr(Z_i = 1) = 1/C_i$ and $C_{i+1} = (1 + Z_i)C_i$.

If we fix C_i then $\mathbb{E}(1 + Z_i)C_i = (1 + \frac{1}{C_i})C_i = C_i + 1$.

Now take expectations of both sides: $\mathbb{E}(C_{i+1}) = \mathbb{E}(C_i) + 1$

MORRIS - Analysis of Expectation

Let us look at the first two symbols that arrive.

- After the first token, $\mathbb{E}(C_1) = 2$. The variable c will then be set to 2.
- After the second token, $\mathbb{E}(C_2) = \frac{1}{2} \cdot 2^2 + \frac{1}{2}2 = 3$. The variable c will be set to 4 with probability $\frac{1}{2}$.

LEMMA (Expectation of Morris's algorithm)

For all $n \geq 0$, $\mathbb{E}(C_n) = n + 1$

Proof.

Let r.v. $Z_i = 1$ if c is increased when the ith symbol arrives and 0 otherwise.

$\Pr(Z_i = 1) = 1/C_i$ and $C_{i+1} = (1 + Z_i)C_i$.

If we fix C_i then $\mathbb{E}(1 + Z_i)C_i = (1 + \frac{1}{C_i})C_i = C_i + 1$.

Now take expectations of both sides: $\mathbb{E}(C_{i+1}) = \mathbb{E}(C_i) + 1$

Therefore $\mathbb{E}(C_n) = n + 1$ since $\mathbb{E}(C_0) = 1$. □

Morris - Analysis of Variance

Lemma (Variance of Morris's algorithm)

For all $n \geq 0$, $\text{var}(C_n) = n(n-1)/2$

Proof.

MORRIS - Analysis of Variance

LEMMA (Variance of Morris's algorithm)

For all $n \geq 0$, $\mathrm{var}(C_n) = n(n-1)/2$

Proof.

$$C_{i+1}^2 = (1 + 2Z_i + Z_i^2)C_i^2 = (1 + 3Z_i)C_i^2$$

MORRIS - Analysis of Variance

LEMMA (Variance of Morris's algorithm)

For all $n \geq 0$, $\mathrm{var}(C_n) = n(n-1)/2$

Proof.

$C_{i+1}^2 = (1 + 2Z_i + Z_i^2)C_i^2 = (1 + 3Z_i)C_i^2$

If we fix C_i then $\mathbb{E}((1 + 3Z_i)C_i^2) = (1 + \frac{3}{C_i})C_i^2 = C_i^2 + 3C_i$.

MORRIS - Analysis of Variance

LEMMA (Variance of Morris's algorithm)

For all $n \geq 0$, $\text{var}(C_n) = n(n-1)/2$

Proof.

$C_{i+1}^2 = (1 + 2Z_i + Z_i^2)C_i^2 = (1 + 3Z_i)C_i^2$

If we fix C_i then $\mathbb{E}((1 + 3Z_i)C_i^2) = (1 + \frac{3}{C_i})C_i^2 = C_i^2 + 3C_i$.

Therefore $\mathbb{E}(C_{i+1}^2) = \mathbb{E}(C_i^2) + 3\mathbb{E}(C_i) = \mathbb{E}(C_i^2) + 3(i+1)$.

MORRIS - Analysis of Variance

LEMMA (Variance of Morris's algorithm)

For all $n \geq 0$, $\mathrm{var}(C_n) = n(n-1)/2$

Proof.

$C_{i+1}^2 = (1 + 2Z_i + Z_i^2)C_i^2 = (1 + 3Z_i)C_i^2$

If we fix C_i then $\mathbb{E}((1 + 3Z_i)C_i^2) = (1 + \frac{3}{C_i})C_i^2 = C_i^2 + 3C_i$.

Therefore $\mathbb{E}(C_{i+1}^2) = \mathbb{E}(C_i^2) + 3\mathbb{E}(C_i) = \mathbb{E}(C_i^2) + 3(i+1)$.

Since $\mathbb{E}(C_0^2) = 1$ we have $\mathbb{E}(C_n^2) = 1 + \frac{3n(n+1)}{2}$.

MORRIS - Analysis of Variance

LEMMA (Variance of Morris's algorithm)

For all $n \geq 0$, $\mathrm{var}(C_n) = n(n-1)/2$

Proof.

$C_{i+1}^2 = (1 + 2Z_i + Z_i^2)C_i^2 = (1 + 3Z_i)C_i^2$

If we fix C_i then $\mathbb{E}((1 + 3Z_i)C_i^2) = (1 + \frac{3}{C_i})C_i^2 = C_i^2 + 3C_i$.

Therefore $\mathbb{E}(C_{i+1}^2) = \mathbb{E}(C_i^2) + 3\mathbb{E}(C_i) = \mathbb{E}(C_i^2) + 3(i+1)$.

Since $\mathbb{E}(C_0^2) = 1$ we have $\mathbb{E}(C_n^2) = 1 + \frac{3n(n+1)}{2}$.

Finally, $\mathrm{var}(C_n) = \mathbb{E}(C_n^2) - (\mathbb{E}(C_n))^2 = \frac{n(n-1)}{2}$ □

Morris - Median of Means

We would like to make it less likely that our estimate is a long way off. It won't work to take the median of k independent runs as we did for Tidemark because the variance of our estimator is too large.

Morris - Median of Means

We would like to make it less likely that our estimate is a long way off. It won't work to take the median of k independent runs as we did for Tidemark because the variance of our estimator is too large.

What can we do?

Morris - Median of Means

We would like to make it less likely that our estimate is a long way off. It won't work to take the median of k independent runs as we did for Tidemark because the variance of our estimator is too large.

What can we do?

Take the mean to reduce the variance, then take the median.

Morris - Median of Means

We would like to make it less likely that our estimate is a long way off. It won't work to take the median of k independent runs as we did for Tidemark because the variance of our estimator is too large.

What can we do?

Take the mean to reduce the variance, then take the median.

Repeat t times: each time take the mean of k independent runs.

MORRIS - Median of Means

We would like to make it less likely that our estimate is a long way off. It won't work to take the median of k independent runs as we did for TIDEMARK because the variance of our estimator is too large.

What can we do?

Take the mean to reduce the variance, then take the median.

Repeat t times: each time take the mean of k independent runs.

Take the median of these t values.

Morris - Median of Means

We would like to make it less likely that our estimate is a long way off. It won't work to take the median of k independent runs as we did for Tidemark because the variance of our estimator is too large.

What can we do?

Take the mean to reduce the variance, then take the median.

Repeat t times: each time take the mean of k independent runs.

Take the median of these t values.

Return this median as our estimate.

MORRIS - Median of Means

We would like to make it less likely that our estimate is a long way off. It won't work to take the median of k independent runs as we did for TIDEMARK because the variance of our estimator is too large.

What can we do?

Take the mean to reduce the variance, then take the median.

Repeat t times: each time take the mean of k independent runs.

Take the median of these t values.

Return this median as our estimate.

This estimate will be much less likely to be bad.

MORRIS - The main result Ia

Repeat t iterations of k independent runs. Let $X_{i,j}$ be unbiased estimators for the count whose true value we call Q. Let X be distributed identically to $X_{i,j}$. For $\delta, \epsilon > 0$, set

$$t = c \left\lceil \log_2 \frac{1}{\delta} \right\rceil$$

$$k = \frac{3\,\mathrm{var}(X)}{\epsilon^2 (\mathbb{E}(X))^2}$$

MORRIS - The main result Ia

Repeat t iterations of k independent runs. Let $X_{i,j}$ be unbiased estimators for the count whose true value we call Q. Let X be distributed identically to $X_{i,j}$. For $\delta, \epsilon > 0$, set

$$t = c \left\lceil \log_2 \frac{1}{\delta} \right\rceil$$

$$k = \frac{3\,\mathrm{var}(X)}{\epsilon^2(\mathbb{E}(X))^2}$$

Let $Z = \mathrm{median}_{i \in [t]} \left(\underbrace{\frac{1}{k} \sum_{j=1}^{k} X_{i,j}}_{mean} \right)$.

Morris - The main result Ia

Repeat t iterations of k independent runs. Let $X_{i,j}$ be unbiased estimators for the count whose true value we call Q. Let X be distributed identically to $X_{i,j}$. For $\delta, \epsilon > 0$, set

$$t = c \left\lceil \log_2 \frac{1}{\delta} \right\rceil$$

$$k = \frac{3\,\mathrm{var}(X)}{\epsilon^2 (\mathbb{E}(X))^2}$$

Let $Z = \mathrm{median}_{i \in [t]} \left(\underbrace{\frac{1}{k} \sum_{j=1}^{k} X_{i,j}}_{mean} \right)$. Then we have that

$\Pr(|Z - Q| \geq \epsilon Q) \leq \delta$. That is Z is an (ϵ, δ)-estimate of Q.

$\textsc{Morris}$ - The main result Ia

Repeat t iterations of k independent runs. Let $X_{i,j}$ be unbiased estimators for the count whose true value we call Q. Let X be distributed identically to $X_{i,j}$. For $\delta, \epsilon > 0$, set

$$t = c \left\lceil \log_2 \frac{1}{\delta} \right\rceil$$

$$k = \frac{3\,\mathrm{var}(X)}{\epsilon^2(\mathbb{E}(X))^2}$$

Let $Z = \mathrm{median}_{i \in [t]} \left(\underbrace{\frac{1}{k} \sum_{j=1}^{k} X_{i,j}}_{mean} \right)$. Then we have that

$\Pr(|Z - Q| \geq \epsilon Q) \leq \delta$. That is Z is an (ϵ, δ)-estimate of Q.
If $\textsc{Morris}$ uses s bits then our (ϵ, δ)-estimate uses

$$O\left(s \cdot \frac{\mathrm{var}(X)}{(\mathbb{E}(X))^2} \cdot \frac{1}{\epsilon^2} \cdot \log \frac{1}{\delta} \right) \text{ bits.}$$

Morris - The main result Ib

Lemma (Preliminary (ϵ, δ) result)

$\Pr(|Z - Q| \geq \epsilon Q) \leq \delta$. That is Z is an (ϵ, δ)-estimate of Q.

Proof.

□

MORRIS - The main result Ib

LEMMA (Preliminary (ϵ, δ) result)

$\Pr(|Z - Q| \geq \epsilon Q) \leq \delta$. That is Z is an (ϵ, δ)-estimate of Q.

Proof.

For each $i \in [t]$ we know $\mathbb{E}(\frac{1}{k} \cdot \sum_{j=1}^{k} X_{i,j}) = Q$ by linearity of expectation.

□

Morris - The main result Ib

Lemma (Preliminary (ϵ, δ) result)

$\Pr(|Z - Q| \geq \epsilon Q) \leq \delta$. That is Z is an (ϵ, δ)-estimate of Q.

Proof.

For each $i \in [t]$ we know $\mathbb{E}(\frac{1}{k} \cdot \sum_{j=1}^{k} X_{i,j}) = Q$ by linearity of expectation.

From pairwise independence, $\mathrm{var}(\frac{1}{k^2} \cdot \sum_{j=1}^{k} X_{i,j}) = \frac{\mathrm{var}(X)}{k}$.

□

MORRIS - The main result Ib

LEMMA (Preliminary (ϵ, δ) result)

$\Pr(|Z - Q| \geq \epsilon Q) \leq \delta$. That is Z is an (ϵ, δ)-estimate of Q.

Proof.

For each $i \in [t]$ we know $\mathbb{E}(\frac{1}{k} \cdot \sum_{j=1}^{k} X_{i,j}) = Q$ by linearity of expectation.

From pairwise independence, $\mathrm{var}(\frac{1}{k^2} \cdot \sum_{j=1}^{k} X_{i,j}) = \frac{\mathrm{var}(X)}{k}$.

Let $Y_i = \frac{1}{k^2} \cdot \sum_{j=1}^{k} X_{i,j}$,

$$\Pr(|Y_i - Q| \geq \epsilon Q) \leq \frac{\mathrm{var}(Y_i)}{(\epsilon Q)^2} = \frac{\mathrm{var}(X)}{k\epsilon^2(\mathbb{E}(X))^2} = \frac{1}{3}$$

□

MORRIS - The main result Ib

LEMMA (Preliminary (ϵ, δ) result

$\Pr(|Z - Q| \geq \epsilon Q) \leq \delta$. That is Z is an (ϵ, δ)-estimate of Q.

Proof.

For each $i \in [t]$ we know $\mathbb{E}(\frac{1}{k} \cdot \sum_{j=1}^{k} X_{i,j}) = Q$ by linearity of expectation.

From pairwise independence, $\mathrm{var}(\frac{1}{k^2} \cdot \sum_{j=1}^{k} X_{i,j}) = \frac{\mathrm{var}(X)}{k}$.

Let $Y_i = \frac{1}{k^2} \cdot \sum_{j=1}^{k} X_{i,j}$,

$$\Pr(|Y_i - Q| \geq \epsilon Q) \leq \frac{\mathrm{var}(Y_i)}{(\epsilon Q)^2} = \frac{\mathrm{var}(X)}{k\epsilon^2(\mathbb{E}(X))^2} = \frac{1}{3}$$

Now apply the median trick from Lecture 4 (TIDEMARK) to get the desired result.

□

Morris - The main result II

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log\log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Proof.

MORRIS - The main result II

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log\log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Proof.

We know that $\frac{\mathrm{var}(X)}{(\mathbb{E}(X))^2} = \frac{n(n-1)}{2n^2} = \frac{1}{2} - \frac{1}{2n}$. Therefore the estimator uses $O(s \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

MORRIS - The main result II

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log\log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Proof.

We know that $\frac{\mathrm{var}(X)}{(\mathbb{E}(X))^2} = \frac{n(n-1)}{2n^2} = \frac{1}{2} - \frac{1}{2n}$. Therefore the estimator uses $O(s \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Set a maximum $s = 1 + \log_2 \log_2 m$ by aborting if the variable x is greater than $2 \log_2 m$.

MORRIS - The main result II

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Proof.

We know that $\frac{\text{var}(X)}{(\mathbb{E}(X))^2} = \frac{n(n-1)}{2n^2} = \frac{1}{2} - \frac{1}{2n}$. Therefore the estimator uses $O(s \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Set a maximum $s = 1 + \log_2 \log_2 m$ by aborting if the variable x is greater than $2 \log_2 m$.

This implies that $C_m \geq m^2 \geq n^2$. Therefore

$$\Pr(C_n \geq n^2) \leq \frac{\mathbb{E}(C_n)}{n^2} = \frac{n+1}{n^2} = \frac{1}{n} + \frac{1}{n^2}$$

MORRIS - The main result II

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Proof.

We know that $\frac{\text{var}(X)}{(\mathbb{E}(X))^2} = \frac{n(n-1)}{2n^2} = \frac{1}{2} - \frac{1}{2n}$. Therefore the estimator uses $O(s \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Set a maximum $s = 1 + \log_2 \log_2 m$ by aborting if the variable x is greater than $2 \log_2 m$.

This implies that $C_m \geq m^2 \geq n^2$. Therefore

$$\Pr(C_n \geq n^2) \leq \frac{\mathbb{E}(C_n)}{n^2} = \frac{n+1}{n^2} = \frac{1}{n} + \frac{1}{n^2}$$

The probability that any one of the $O(\epsilon^{-2} \log \delta^{-1})$ runs aborts is $o(1)$. (Union bound)

MORRIS - The main result III

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Is this any good in practice? Think of $\epsilon = 1/2$.

MORRIS - The main result III

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Is this any good in practice? Think of $\epsilon = 1/2$.

Exercise 4-1 shows how to improve the result to $O(\log \log m + \log \epsilon^{-1} + \log \delta^{-1})$ bits which is a significant improvement.

MORRIS - The main result III

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Is this any good in practice? Think of $\epsilon = 1/2$.

Exercise 4-1 shows how to improve the result to $O(\log \log m + \log \epsilon^{-1} + \log \delta^{-1})$ bits which is a significant improvement.

In practice you use this if you don't need a very accurate estimate and/or with multiple massive streams.

MORRIS - The main result III

Theorem - Approximate Counting

For a stream of length at most m, the problem of approximately counting the number of tokens admits an (ϵ, δ)-estimation in $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

Is this any good in practice? Think of $\epsilon = 1/2$.

Exercise 4-1 shows how to improve the result to $O(\log \log m + \log \epsilon^{-1} + \log \delta^{-1})$ bits which is a significant improvement.

In practice you use this if you don't need a very accurate estimate and/or with multiple massive streams.

The theory is however very attractive.

Morris - summary

- The Morris algorithms run in $O(m)$ time but is an *unbiased* estimator but the variance is high.

MORRIS - summary

- The MORRIS algorithms run in $O(m)$ time but is an *unbiased* estimator but the variance is high.

- We can improve it by taking the median of means to give $O(\epsilon^{-2} \log \delta^{-1} m)$ time and $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

MORRIS - summary

- The MORRIS algorithms run in $O(m)$ time but is an *unbiased* estimator but the variance is high.

- We can improve it by taking the median of means to give $O(\epsilon^{-2} \log \delta^{-1} m)$ time and $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

- It is one-pass.

MORRIS - summary

- The MORRIS algorithms run in $O(m)$ time but is an *unbiased* estimator but the variance is high.
- We can improve it by taking the median of means to give $O(\epsilon^{-2} \log \delta^{-1} m)$ time and $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.
- It is one-pass.
- The accuracy depends on the choice of ϵ and δ. The smaller they are, the more accurate is the estimate but the longer the algorithms takes to run and the more space it takes.

MORRIS - summary

- The MORRIS algorithms run in $O(m)$ time but is an *unbiased* estimator but the variance is high.

- We can improve it by taking the median of means to give $O(\epsilon^{-2} \log \delta^{-1} m)$ time and $O(\log \log m \cdot \epsilon^{-2} \log \delta^{-1})$ bits of space.

- It is one-pass.

- The accuracy depends on the choice of ϵ and δ. The smaller they are, the more accurate is the estimate but the longer the algorithms takes to run and the more space it takes.

- Exercise 4-1 shows how to improve the space usage to $O(\log \log m + \log \epsilon^{-1} + \log \delta^{-1})$ bits.