# First order optimization.

$\min f(x)$

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."
$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."
$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.
No Lipschitz condition. Works for subgradients.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."
$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.
No Lipschitz condition. Works for subgradients.
Idea: average lower bound is average of linear lower bounds.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."
$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.
No Lipschitz condition. Works for subgradients.
Idea: average lower bound is average of linear lower bounds.
$R(u) - w(x) = \sum_i (\nabla f(x_t))(x - u) - w(u)$.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."
$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.
No Lipschitz condition. Works for subgradients.
Idea: average lower bound is average of linear lower bounds.
$R(u) - w(x) = \sum_i (\nabla f(x_t))(x - u) - w(u)$.
What is $w(x)$?

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."
$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.
No Lipschitz condition. Works for subgradients.
Idea: average lower bound is average of linear lower bounds.
$R(u) - w(x) = \sum_i (\nabla f(x_t))(x - u) - w(u)$.
What is $w(x)$? One option: Euclidean norm of $x$.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."
$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.
No Lipschitz condition. Works for subgradients.
Idea: average lower bound is average of linear lower bounds.
$R(u) - w(x) = \sum_i (\nabla f(x_t))(x - u) - w(u)$.
What is $w(x)$? One option: Euclidean norm of $x$.
Another, $w(x) = \sum_i x_i \log x_i$.

# First order optimization.

$\min f(x)$

Convexity: $f(x) + (\nabla f(x)) \cdot (y - x) \leq f(y)$.
Lipschitz: $\|\nabla(f(x)) - \nabla(f(y))\| \leq L\|x - y\|$
$\nabla f(x)$ - gradient or subgradient.

Gradient Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$
One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:
$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"
Output: Average point.
One bound: Total Difference from optimal or "regret."
$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.
No Lipschitz condition. Works for subgradients.
Idea: average lower bound is average of linear lower bounds.
$R(u) - w(x) = \sum_i (\nabla f(x_t))(x - u) - w(u)$.
What is $w(x)$? One option: Euclidean norm of $x$.
Another, $w(x) = \sum_i x_i \log x_i$. Get multiplicative weight update!!!!

# Last time.

Gradient Descent:

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$.

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

“Mirror” Descent:

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

“Mirror” Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for “euclidean proximity function”

Output: Average point.

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

Accelerated Gradient Descent:

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

Accelerated Gradient Descent:

$x_{t+1} = x + \alpha_i (x_t - x_{t-1}) - \beta_i \nabla f(x_t)$.

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

Accelerated Gradient Descent:

$x_{t+1} = x + \alpha_i (x_t - x_{t-1}) - \beta_i \nabla f(x_t)$.

Momentum term: $(x_t - x_{t+1}) = \sum_i \nu_i \nabla (f_( x_i))$.

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

Accelerated Gradient Descent:

$x_{t+1} = x + \alpha_i (x_t - x_{t-1}) - \beta_i \nabla f(x_t)$.

Momentum term: $(x_t - x_{t+1}) = \sum_i \nu_i \nabla (f_( x_i))$.

where $\sum_i \nu_i = 1$.

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

Accelerated Gradient Descent:

$x_{t+1} = x + \alpha_i (x_t - x_{t-1}) - \beta_i \nabla f(x_t)$.

Momentum term: $(x_t - x_{t+1}) = \sum_i \nu_i \nabla(f_(x_i))$.

where $\sum_i \nu_i = 1$.

Mirror Descent point!

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

Accelerated Gradient Descent:

$x_{t+1} = x + \alpha_i(x_t - x_{t-1}) - \beta_i \nabla f(x_t)$.

Momentum term: $(x_t - x_{t+1}) = \sum_i \nu_i \nabla(f_( x_i))$.

where $\sum_i \nu_i = 1$.

Mirror Descent point!

Idea of Analysis:

# Last time.

Gradient Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$

One bound: $f(x_t) - f(x_{t+1}) \geq \frac{\|\nabla f(x_t)\|^2}{L}$. Lipschitz.

"Mirror" Descent:

$x_{t+1} = x_t - \alpha \nabla f(x_t)$ for "euclidean proximity function"

Output: Average point.

One bound: Total Difference from optimal or "regret."

$\sum_t \alpha \|\nabla f(x_t)\|^2 + \frac{w(u)}{T}$.

Accelerated Gradient Descent:

$x_{t+1} = x + \alpha_i (x_t - x_{t-1}) - \beta_i \nabla f(x_t)$.

Momentum term: $(x_t - x_{t+1}) = \sum_i \nu_i \nabla(f_(x_i))$.

where $\sum_i \nu_i = 1$.

Mirror Descent point!

Idea of Analysis:

Benefit for gradient cancels some of regret term of MD.

# Other scenarios.

Don’t you dual norm me!

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$.

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

For Euclidean norm, what is dual norm?

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?
$\|x\|_1 = \sum_i |x_i|$.

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1}\langle x, y\rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?
$\|x\|_1 = \sum_i |x_i|$.
$\|x\|_\infty = \max_i |x_i|$.

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?
$\|x\|_1 = \sum_i |x_i|$.
$\|x\|_\infty = \max_i |x_i|$.

Can be Lipschitz in different norms:

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?
$\|x\|_1 = \sum_i |x_i|$.
$\|x\|_\infty = \max_i |x_i|$.

Can be Lipschitz in different norms:
$\|\nabla f(x) - \nabla f(y)\|_* = L\|x - y\|$.

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1}\langle x, y\rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?
$\|x\|_1 = \sum_i |x_i|$.
$\|x\|_\infty = \max_i |x_i|$.

Can be Lipschitz in different norms:
$\|\nabla f(x) - \nabla f(y)\|_* = L\|x - y\|$.

Gradient Step:

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?
$\|x\|_1 = \sum_i |x_i|$.
$\|x\|_\infty = \max_i |x_i|$.

Can be Lipschitz in different norms:
$\|\nabla f(x) - \nabla f(y)\|_* = L\|x - y\|$.

Gradient Step:
$x_{t+1} = x_t - \alpha argmax_{|y|=1} \langle \nabla(f(x)), y \rangle$.

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?
$\|x\|_1 = \sum_i |x_i|$.
$\|x\|_\infty = \max_i |x_i|$.

Can be Lipschitz in different norms:
$\|\nabla f(x) - \nabla f(y)\|_* = L\|x - y\|$.

Gradient Step:
$x_{t+1} = x_t - \alpha argmax_{|y|=1} \langle \nabla (f(x)), y \rangle$.

Lipschitz in $\ell_1$, when optimizing $\sum_i |x_i|$.

# Other scenarios.

Don't you dual norm me!

Norm: $\|x\|$. Dual Norm: $\|y\|_*$.

$\|y\|_* = \max_{\|x\|=1} \langle x, y \rangle$.

For Euclidean norm, what is dual norm?

For $\ell_1$ or hamming norm, what is dual norm?
$\|x\|_1 = \sum_i |x_i|$.
$\|x\|_\infty = \max_i |x_i|$.

Can be Lipschitz in different norms:
$\|\nabla f(x) - \nabla f(y)\|_* = L\|x - y\|$.

Gradient Step:
$x_{t+1} = x_t - \alpha argmax_{|y|=1} \langle \nabla(f(x)), y \rangle$.

Lipschitz in $\ell_1$, when optimizing $\sum_i |x_i|$.
E.g. Max Flow or tolls.

# Next Topic

Streaming.

# Next Topic

Streaming.

Frequent Items.

# Streaming

Stream: $x_1$,

# Streaming

Stream: $x_1, x_2,$

# Streaming

Stream: $x_1, x_2, x_3,$

# Streaming

Stream: $x_1, x_2, x_3, , \ldots x_n$

# Streaming

Stream: $x_1, x_2, x_3, , \ldots x_n$

Resources: $O(\log^c n)$ storage.

# Streaming

Stream: $x_1, x_2, x_3, , \ldots x_n$

Resources: $O(\log^c n)$ storage.

Today's Goal: find frequent items.

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k + 1$th item?

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k+1$th item?

Yes?

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k+1$th item?

Yes?
No?

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k+1$th item?

Yes?
No?
$k+1$st most frequent item occurs $< \frac{n}{k+1}$

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k + 1$th item?

Yes?
No?
$k + 1$st most frequent item occurs $< \frac{n}{k+1}$
  Off by 100%.

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k + 1$th item?

Yes?
No?
$k + 1$st most frequent item occurs $< \frac{n}{k+1}$
Off by 100%. 0 estimate is fine.

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k+1$th item?

Yes?
No?
$k+1$st most frequent item occurs $< \frac{n}{k+1}$
Off by 100%. 0 estimate is fine.

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k+1$th item?

Yes?
No?
$k+1$st most frequent item occurs $< \frac{n}{k+1}$
  Off by 100%. 0 estimate is fine.
No item more frequent than $\frac{n}{k}$?

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k+1$th item?

Yes?
No?
$k+1$st most frequent item occurs $< \frac{n}{k+1}$
Off by 100%. 0 estimate is fine.
No item more frequent than $\frac{n}{k}$?
0 estimate is fine.

# Frequent Items: deterministic.

Additive $\frac{n}{k}$ error.

Accurate count for $k+1$th item?

Yes?
No?
$k+1$st most frequent item occurs $< \frac{n}{k+1}$
  Off by 100%. 0 estimate is fine.
No item more frequent than $\frac{n}{k}$?
  0 estimate is fine.

Only reasonable for frequent items.

# Deteministic Algorithm.

Alg:

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$

If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters. Delete zero count elts.

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$

If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters. Delete zero count elts.

Example:

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters. Delete zero count elts.

Example:

State: $k = 3$

Stream

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$
If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters. Delete zero count elts.

Example:

State: $k = 3$

Stream

$[(1, 1)]$

$1,$

Previous State

$[]$

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$

If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters. Delete zero count elts.

Example:

State: $k = 3$

Stream

$[(1,1) - -(2,1)]$

$1, 2$

Previous State
$[(1,1)]$

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters. Delete zero count elts.

Example:

State: $k = 3$

Stream

$$[(1, 1) - -(2, 1) - -(3, 1)]$$

$1, 2, 3$

Previous State
$$[(1, 1) - -(2, 1)]$$

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$

If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters. Delete zero count elts.

Example:

State: $k = 3$

Stream

$[(1,2) - -(2,1) - -(3,1)]$

$1, 2, 3, 1$

Previous State
$[(1,1) - -(2,1) - -(3,1)]$

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters. Delete zero count elts.

Example:

State: $k = 3$

Stream

$[(1,2) - -(2,2) - -(3,1)]$

$1, 2, 3, 1, 2$

Previous State
$[(1,2) - -(2,1) - -(3,1)]$

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters. Delete zero count elts.

Example:

State: $k = 3$

Stream

$[(1, 1) - -(2, 1) - -(3, 0)]$

$1, 2, 3, 1, 2, 4$

Previous State
$[(1, 2) - -(2, 2) - -(3, 1)]$

# Deteministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$

If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters. Delete zero count elts.

Example:

State: $k = 3$

Stream

# Deterministic Algorithm.

Alg:

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

Estimate for item:

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$

If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters.

Estimate for item:
if in $S$, value of counter.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
  If $S$ has space, add $x_i$ to $S$ w/value 1.
  Otherwise decrement all counters.

Estimate for item:
 if in $S$, value of counter.
 otherwise 0.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$

  If $S$ has space, add $x_i$ to $S$ w/value 1.
  Otherwise decrement all counters.

Estimate for item:
 if in $S$, value of counter.
 otherwise 0.
Underestimate

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$

  If $S$ has space, add $x_i$ to $S$ w/value 1.
  Otherwise decrement all counters.

Estimate for item:
  if in $S$, value of counter.
  otherwise 0.
Underestimate clearly.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

Estimate for item:
  if in $S$, value of counter.
  otherwise 0.
Underestimate clearly.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

Estimate for item:
  if in $S$, value of counter.
  otherwise 0.
Underestimate clearly.
  Increment once when see an item, might decrement.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

Estimate for item:
  if in $S$, value of counter.
  otherwise 0.
Underestimate clearly.
  Increment once when see an item, might decrement.
Total decrements, $T$?

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
  If $S$ has space, add $x_i$ to $S$ w/value 1.
  Otherwise decrement all counters.

Estimate for item:
 if in $S$, value of counter.
 otherwise 0.
Underestimate clearly.
 Increment once when see an item, might decrement.
Total decrements, $T$? $n$?

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
  If $S$ has space, add $x_i$ to $S$ w/value 1.
  Otherwise decrement all counters.

Estimate for item:
 if in $S$, value of counter.
 otherwise 0.
Underestimate clearly.
 Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$?

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

Estimate for item:
  if in $S$, value of counter.
  otherwise 0.
Underestimate clearly.
  Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

Estimate for item:
  if in $S$, value of counter.
  otherwise 0.
Underestimate clearly.
  Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

  decrement $k$ counters on each decrement.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

Estimate for item:
  if in $S$, value of counter.
  otherwise 0.
Underestimate clearly.
  Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

  decrement $k$ counters on each decrement.
  $Tk$ total decremting

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
  If $S$ has space, add $x_i$ to $S$ w/value 1.
  Otherwise decrement all counters.

Estimate for item:
 if in $S$, value of counter.
 otherwise 0.
Underestimate clearly.
 Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

 decrement $k$ counters on each decrement.
 $Tk$ total decremting
 $n$ items.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \not\in S$
  If $S$ has space, add $x_i$ to $S$ w/value 1.
  Otherwise decrement all counters.

Estimate for item:
 if in $S$, value of counter.
 otherwise 0.
Underestimate clearly.
 Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

 decrement $k$ counters on each decrement.
 $Tk$ total decremting
 $n$ items. $n$ total incrementing.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters.

Estimate for item:
if in $S$, value of counter.
otherwise 0.
Underestimate clearly.
Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

decrement $k$ counters on each decrement.
$Tk$ total decremting
$n$ items. $n$ total incrementing.
$\implies T \leq \frac{n}{k}$.

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
    If $S$ has space, add $x_i$ to $S$ w/value 1.
    Otherwise decrement all counters.

Estimate for item:
 if in $S$, value of counter.
 otherwise 0.
Underestimate clearly.
  Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

 decrement $k$ counters on each decrement.
  $Tk$ total decremting
 $n$ items. $n$ total incrementing.
$\implies T \leq \frac{n}{k}$.
Off by at most $\frac{n}{k}$

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
If $S$ has space, add $x_i$ to $S$ w/value 1.
Otherwise decrement all counters.

Estimate for item:
if in $S$, value of counter.
otherwise 0.
Underestimate clearly.
Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

decrement $k$ counters on each decrement.
$Tk$ total decremting
$n$ items. $n$ total incrementing.
$\implies T \leq \frac{n}{k}$.
Off by at most $\frac{n}{k}$

Space?

# Deterministic Algorithm.

Alg:
(1) Set, $S$, of $k$ counters, initially 0.
(2) If $x_i \in S$ increment $x_i$'s counter.
(3) If $x_i \notin S$
 If $S$ has space, add $x_i$ to $S$ w/value 1.
 Otherwise decrement all counters.

Estimate for item:
 if in $S$, value of counter.
 otherwise 0.
Underestimate clearly.
 Increment once when see an item, might decrement.
Total decrements, $T$? $n$? $n/k$? $k$?

 decrement $k$ counters on each decrement.
 $Tk$ total decremting
 $n$ items. $n$ total incrementing.
$\implies T \leq \frac{n}{k}$.
Off by at most $\frac{n}{k}$

Space? $O(k \log n)$

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$
item $i$, count $c_i$ (possibly negative.)

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$

item $i$, count $c_i$ (possibly negative.)

Positive total for each item!

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$
item $i$, count $c_i$ (possibly negative.)
Positive total for each item!

Estimate frequency of item: $f_j = \sum c_j$.

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$
item $i$, count $c_i$ (possibly negative.)
Positive total for each item!

Estimate frequency of item: $f_j = \sum c_j$.

$|f|_1 = \sum_j |f_j|$

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$
item $i$, count $c_i$ (possibly negative.)
Positive total for each item!

Estimate frequency of item: $f_j = \sum c_j$.

$|f|_1 = \sum_j |f_j|$ Smaller than $\sum_i |c_i|$.

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$
item $i$, count $c_i$ (possibly negative.)
Positive total for each item!

Estimate frequency of item: $f_j = \sum c_j$.

$|f|_1 = \sum_j |f_j|$    Smaller than $\sum_i |c_i|$.

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$
  item $i$, count $c_i$ (possibly negative.)
Positive total for each item!

Estimate frequency of item: $f_j = \sum c_j$.

$|f|_1 = \sum_j |f_j|$   Smaller than $\sum_i |c_i|$.

Approximation:

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$
item $i$, count $c_i$ (possibly negative.)
Positive total for each item!

Estimate frequency of item: $f_j = \sum c_j$.

$|f|_1 = \sum_j |f_j|$ Smaller than $\sum_i |c_i|$.

Approximation:

Additive $\epsilon |f|_1$ with probability $1 - \delta$

# Turnstile Model and Randomization

Stream: $\ldots, (i, c_i), \ldots$

item $i$, count $c_i$ (possibly negative.)

Positive total for each item!

Estimate frequency of item: $f_j = \sum c_j$.

$|f|_1 = \sum_j |f_j|$ Smaller than $\sum_i |c_i|$.

Approximation:

Additive $\epsilon |f|_1$ with probability $1 - \delta$

Space $O(\frac{1}{\epsilon} \log \frac{1}{\delta} \log n)$.

# Count Min Sketch

Sketch

# Count Min Sketch

Sketch – Summary of stream.

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)]$ **+=** $c_j$.

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)]$ **+=** $c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)]$ **+=** $c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

Intuition: $|f|_1/k$ other "counts" in same bucket.

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)]$ **+=** $c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

Intuition:$|f|_1/k$ other "counts" in same bucket.

$\rightarrow$ Additive $|f|_1/k$ error on average for each of $t$ arrays.

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)]$ **+=** $c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

Intuition:$|f|_1/k$ other "counts" in same bucket.

$\rightarrow$ Additive $|f|_1/k$ error on average for each of $t$ arrays.

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)]$ **+=** $c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

Intuition:$|f|_1/k$ other "counts" in same bucket.

$\rightarrow$ Additive $|f|_1/k$ error on average for each of $t$ arrays.

Why $t$ buckets?

# Count Min Sketch

Sketch – Summary of stream.

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)]$ **+=** $c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

Intuition: $|f|_1/k$ other "counts" in same bucket.

$\rightarrow$ Additive $|f|_1/k$ error on average for each of $t$ arrays.

Why $t$ buckets? To get high probability.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.

(2) Process elt $(j, c_j)$,

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.

(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.

(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.

(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.

(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.

(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X]$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.

(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.

(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

Pr[$X > 2\frac{|f|_1}{k}$ in all t trials]

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

Pr[$X > 2\frac{|f|_1}{k}$ in all t trials] $\leq (\frac{1}{2})^t$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

$\Pr[X > 2\frac{|f|_1}{k}$ in all t trials$] \leq (\frac{1}{2})^t$
$\leq \delta$ when $t = \log \frac{1}{\delta}$.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
  $h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
  $A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
 $X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
 Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

Pr[$X > 2\frac{|f|_1}{k}$ in all t trials] $\leq (\frac{1}{2})^t$
 $\leq \delta$ when $t = \log \frac{1}{\delta}$.

Error $\epsilon |f|_1$ if

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

$\Pr[X > 2\frac{|f|_1}{k}$ in all t trials$] \leq (\frac{1}{2})^t$
$\leq \delta$ when $t = \log \frac{1}{\delta}$.

Error $\epsilon |f|_1$ if $\epsilon = \frac{2}{k}$.

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

Pr[$X > 2\frac{|f|_1}{k}$ in all t trials] $\leq (\frac{1}{2})^t$
$\leq \delta$ when $t = \log \frac{1}{\delta}$.

Error $\epsilon |f|_1$ if $\epsilon = \frac{2}{k}$.

Space?

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

$\Pr[X > 2\frac{|f|_1}{k}$ in all t trials$] \leq (\frac{1}{2})^t$
$\leq \delta$ when $t = \log \frac{1}{\delta}$.

Error $\epsilon|f|_1$ if $\epsilon = \frac{2}{k}$.

Space? $O(k$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
  $h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
  $A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
  $X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
  Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

Pr[$X > 2\frac{|f|_1}{k}$ in all t trials] $\leq (\frac{1}{2})^t$
  $\leq \delta$ when $t = \log \frac{1}{\delta}$.

Error $\epsilon |f|_1$ if $\epsilon = \frac{2}{k}$.

Space? $O(k \log \frac{1}{\delta}$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

Pr[$X > 2\frac{|f|_1}{k}$ in all t trials] $\leq (\frac{1}{2})^t$
$\leq \delta$ when $t = \log \frac{1}{\delta}$.

Error $\epsilon |f|_1$ if $\epsilon = \frac{2}{k}$.

Space? $O(k \log \frac{1}{\delta} \log n)$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
  $h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
  $A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
 $X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
 Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

$\Pr[X > 2\frac{|f|_1}{k}$ in all t trials$] \leq (\frac{1}{2})^t$
 $\leq \delta$ when $t = \log \frac{1}{\delta}$.

Error $\epsilon |f|_1$ if $\epsilon = \frac{2}{k}$.

Space? $O(k \log \frac{1}{\delta} \log n)$  $O(\frac{1}{\epsilon} \log \frac{1}{\delta}$

# Count min sketch:analysis

(1) $t$ arrays, $A[i]$, of $k$ counters.
$h_1, \ldots, h_t$ from 2-wise ind. family.
(2) Process elt $(j, c_j)$,
$A[i][h_i(j)] += c_j$.
(3) Item $j$ estimate: $\min_i A[i][h_i(j)]$.

$A[1][h_j(j)] = f_j + X$, where $X$ is a random variable.

$Y_i$ - item $h_1(i) = h_1(j)$
$X = \sum_i Y_i f_i$

$E[X] = \sum_i E[Y_i] f_i = \sum_i \frac{1}{k} f_i = \frac{|f|_1}{k}$

Markov: $Pr[X > 2\frac{|f|_1}{k}] \leq \frac{1}{2}$
Exercise: proof of Markov. (All above average?)

$t$ independent trials, pick smallest.

$\Pr[X > 2\frac{|f|_1}{k}$ in all t trials$] \leq (\frac{1}{2})^t$
$\leq \delta$ when $t = \log \frac{1}{\delta}$.

Error $\epsilon |f|_1$ if $\epsilon = \frac{2}{k}$.

Space? $O(k \log \frac{1}{\delta} \log n)$ $O(\frac{1}{\epsilon} \log \frac{1}{\delta} \log n)$

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better.

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
$t$ hash functions $h_i : U \rightarrow [k]$

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
$t$ hash functions $h_i : U \rightarrow [k]$
$t$ hash functions $g_i : U \rightarrow [-1, +1]$

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \le |f|_2 \le |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
  $t$ hash functions $h_i : U \to [k]$
  $t$ hash functions $g_i : U \to [-1, +1]$
(2) Elt $(j, c_j)$

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
$t$ hash functions $h_i : U \rightarrow [k]$
$t$ hash functions $g_i : U \rightarrow [-1, +1]$
(2) Elt $(j, c_j)$
$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
    $t$ hash functions $h_i : U \to [k]$
    $t$ hash functions $g_i : U \to [-1, +1]$
(2) Elt $(j, c_j)$
    $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
$t$ hash functions $h_i : U \rightarrow [k]$
$t$ hash functions $g_i : U \rightarrow [-1, +1]$
(2) Elt $(j, c_j)$
$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \le |f|_2 \le |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
  $t$ hash functions $h_i : U \to [k]$
  $t$ hash functions $g_i : U \to [-1, +1]$
(2) Elt $(j, c_j)$
  $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
    $t$ hash functions $h_i : U \to [k]$
    $t$ hash functions $g_i : U \to [-1, +1]$
(2) Elt $(j, c_j)$
    $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

Other items cancel each other out!

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
    $t$ hash functions $h_i : U \rightarrow [k]$
    $t$ hash functions $g_i : U \rightarrow [-1, +1]$
(2) Elt $(j, c_j)$
    $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

Other items cancel each other out!
Tight!

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
  $t$ hash functions $h_i : U \to [k]$
  $t$ hash functions $g_i : U \to [-1, +1]$
(2) Elt $(j, c_j)$
  $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

Other items cancel each other out!
Tight! (Not an asymptotic statement.)

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
   $t$ hash functions $h_i : U \rightarrow [k]$
   $t$ hash functions $g_i : U \rightarrow [-1, +1]$
(2) Elt $(j, c_j)$
   $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

Other items cancel each other out!
Tight! (Not an asymptotic statement.)

Do $t$ times and average?

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
$t$ hash functions $h_i : U \rightarrow [k]$
$t$ hash functions $g_i : U \rightarrow [-1, +1]$
(2) Elt $(j, c_j)$
$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

Other items cancel each other out!
Tight! (Not an asymptotic statement.)

Do $t$ times and average?

No!

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
    $t$ hash functions $h_i : U \rightarrow [k]$
    $t$ hash functions $g_i : U \rightarrow [-1, +1]$
(2) Elt $(j, c_j)$
    $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

Other items cancel each other out!
Tight! (Not an asymptotic statement.)

Do $t$ times and average?

No! Median!

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
    $t$ hash functions $h_i : U \rightarrow [k]$
    $t$ hash functions $g_i : U \rightarrow [-1, +1]$
(2) Elt $(j, c_j)$
    $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

Other items cancel each other out!
Tight! (Not an asymptotic statement.)

Do $t$ times and average?

No! Median! Two ideas!

# Count sketch.

Error in terms of $|f|_2 = \sqrt{\sum_i f_2^2}$.

$\frac{|f|_1}{\sqrt{n}} \leq |f|_2 \leq |f|_1$.

Could be much better. E.g., uniform frequency $\frac{|f|_1}{\sqrt{n}} = |f|_2$

Alg:

(1) $t$ arrays, $A[i]$:
    $t$ hash functions $h_i : U \to [k]$
    $t$ hash functions $g_i : U \to [-1, +1]$
(2) Elt $(j, c_j)$
    $A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Buckets contains signed count (estimate cancels sign.)

Other items cancel each other out!
Tight! (Not an asymptotic statement.)

Do $t$ times and average?

No! Median! Two ideas! One simple algorithm!

# Analysis

(1) $\cdots$

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$

(2) Elt $(j, c_j)$

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$

(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

# Analysis

(1) $\cdots\ g_i : U \rightarrow [-1, +1], h_i : U \rightarrow [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$
$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

# Analysis

(1) $\cdots\ g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice:

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

# Analysis

(1) $\cdots\ g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$

# Analysis

(1) $\cdots\ g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise

# Analysis

(1) $\cdots\ g_i : U \rightarrow [-1, +1], h_i : U \rightarrow [k]$
(2) Elt $(j, c_j)$
$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$

# Analysis

(1) $\cdots\ g_i : U \to [-1,+1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

# Analysis

(1) $\cdots\ g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$
$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$
(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!

$Var[X] = \sum_{i \in [m]} Var(Y_i)$

# Analysis

(1) $\cdots$ $g_i : U \rightarrow [-1, +1], h_i : U \rightarrow [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$

$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise

$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!

$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k}$

# Analysis

(1) $\cdots$ $g_i : U \to [-1,+1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$

$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise

$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!

$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

# Analysis

(1) $\cdots$ $g_i : U \rightarrow [-1, +1], h_i : U \rightarrow [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$

# Analysis

(1) $\cdots$ $g_i : U \rightarrow [-1, +1], h_i : U \rightarrow [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$

$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise

$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!

$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$

Choose $k = \frac{4}{\epsilon^2}$:

# Analysis

(1) $\cdots\ g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$

Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon|f|_2]$

# Analysis

(1) $\cdots\ g_i : U \rightarrow [-1, +1], h_i : U \rightarrow [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$

$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise

$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!

$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$

Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon |f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2 |f|_2^2}$

# Analysis

(1) $\cdots$ $g_i : U \rightarrow [-1, +1], h_i : U \rightarrow [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$
Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon|f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2|f|_2^2} \leq \frac{\epsilon^2|f|_2^2/4}{\epsilon^2|f|_2^2}$

# Analysis

(1) $\cdots$ $g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$

$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise

$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!

$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$

Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon |f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2 |f|_2^2} \leq \frac{\epsilon^2 |f|_2^2/4}{\epsilon^2 |f|_2^2} \leq \frac{1}{4}$.

# Analysis

(1) $\cdots\ g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$
Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon|f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2|f|_2^2} \leq \frac{\epsilon^2|f|_2^2/4}{\epsilon^2|f|_2^2} \leq \frac{1}{4}$.
Each trial is close with probability $3/4$.

# Analysis

(1) $\cdots\ g_i : U \to [-1,+1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$
Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon|f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2|f|_2^2} \leq \frac{\epsilon^2|f|_2^2/4}{\epsilon^2|f|_2^2} \leq \frac{1}{4}$.
Each trial is close with probability $3/4$.
If $>$ half tosses close, median is close!

# Analysis

(1) $\cdots$ $g_i : U \rightarrow [-1, +1], h_i : U \rightarrow [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$
Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon|f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2|f|_2^2} \leq \frac{\epsilon^2|f|_2^2/4}{\epsilon^2|f|_2^2} \leq \frac{1}{4}$.
Each trial is close with probability $3/4$.
If $>$ half tosses close, median is close!
Exists $t = \Theta(\log \frac{1}{\delta})$

# Analysis

(1) $\cdots$ $g_i : U \to [-1,+1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$
Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon |f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2 |f|_2^2} \leq \frac{\epsilon^2 |f|_2^2/4}{\epsilon^2 |f|_2^2} \leq \frac{1}{4}$.
Each trial is close with probability $3/4$.
If $>$ half tosses close, median is close!
Exists $t = \Theta(\log \frac{1}{\delta})$ where $\geq \frac{1}{2}$ are correct with probability $\geq 1 - \delta$

# Analysis

(1) $\cdots\ g_i : U \to [-1, +1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$
Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon |f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2 |f|_2^2} \leq \frac{\epsilon^2 |f|_2^2/4}{\epsilon^2 |f|_2^2} \leq \frac{1}{4}$.
Each trial is close with probability $3/4$.
If $>$ half tosses close, median is close!
Exists $t = \Theta(\log \frac{1}{\delta})$ where $\geq \frac{1}{2}$ are correct with probability $\geq 1 - \delta$
Total Space: $O(\frac{\log \frac{1}{\delta}}{\epsilon^2}$

# Analysis

(1) $\cdots$ $g_i : U \to [-1,+1], h_i : U \to [k]$
(2) Elt $(j, c_j)$

$$A[i][h(j)] = A[i][h_i(j)] + g_i(j)c_j$$

(3) Item $j$ estimate: median of $g_i(j)A[i][h_i(j)]$.

Notice: $A[1][h_1(j)] = g_1(j)f_j + X$

$X = \sum_i Y_i$
$Y_i = \pm f_i$ if item $h_1(i) = h_1(j)$ $Y_i = 0$, otherwise
$E[Y_i] = 0$ $Var(Y_i) = \frac{f_i^2}{k}$.

$E[X] = 0$ Expected drift is 0!
$Var[X] = \sum_{i \in [m]} Var(Y_i) = \sum_i \frac{f_i^2}{k} = \frac{|f|_2^2}{k}$

Cheybshev: $Pr[|X - \mu| > \Delta] \leq \frac{Var(X)^2}{\Delta^2}$
Choose $k = \frac{4}{\epsilon^2}$: $Pr[|X| > \epsilon|f|_2] \leq \frac{|f|_2^2/k}{\epsilon^2|f|_2^2} \leq \frac{\epsilon^2|f|_2^2/4}{\epsilon^2|f|_2^2} \leq \frac{1}{4}$.
Each trial is close with probability $3/4$.
If $>$ half tosses close, median is close!
Exists $t = \Theta(\log \frac{1}{\delta})$ where $\geq \frac{1}{2}$ are correct with probability $\geq 1 - \delta$
Total Space: $O(\frac{\log \frac{1}{\delta}}{\epsilon^2} \log n)$

# Sum up

Deterministic:

# Sum up

Deterministic:
  stream has items

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:

# Sum up

Deterministic:

stream has items

Count within additive $\frac{n}{k}$

$O(k \log n)$ space.

Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:

stream has $\pm$ counts

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:
stream has $\pm$ counts
Count within additive $\epsilon |f|_1$

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:
stream has $\pm$ counts
Count within additive $\epsilon|f|_1$
with probability at least $1 - \delta$

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:
stream has $\pm$ counts
Count within additive $\epsilon |f|_1$
with probability at least $1 - \delta$
$O(\frac{\log n \log \frac{1}{\delta}}{\epsilon})$.

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:
stream has $\pm$ counts
Count within additive $\epsilon |f|_1$
with probability at least $1 - \delta$
$O(\frac{\log n \log \frac{1}{\delta}}{\epsilon})$.

Count Sketch:

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:
stream has $\pm$ counts
Count within additive $\epsilon |f|_1$
with probability at least $1 - \delta$
$O(\frac{\log n \log \frac{1}{\delta}}{\epsilon})$.

Count Sketch:
stream has $\pm$ counts

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:
stream has $\pm$ counts
Count within additive $\epsilon |f|_1$
with probability at least $1 - \delta$
$O(\frac{\log n \log \frac{1}{\delta}}{\epsilon})$.

Count Sketch:
stream has $\pm$ counts
Count within additive $\epsilon |f|_2$

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:
stream has $\pm$ counts
Count within additive $\epsilon |f|_1$
with probability at least $1 - \delta$
$O(\frac{\log n \log \frac{1}{\delta}}{\epsilon})$.

Count Sketch:
stream has $\pm$ counts
Count within additive $\epsilon |f|_2$
with probability at least $1 - \delta$

# Sum up

Deterministic:
stream has items
Count within additive $\frac{n}{k}$
$O(k \log n)$ space.
Within $\epsilon n$ with $O(\frac{1}{\epsilon} \log n)$ space.

Count Min:
stream has $\pm$ counts
Count within additive $\epsilon|f|_1$
with probability at least $1 - \delta$
$O(\frac{\log n \log \frac{1}{\delta}}{\epsilon})$.

Count Sketch:
stream has $\pm$ counts
Count within additive $\epsilon|f|_2$
with probability at least $1 - \delta$
$O(\frac{\log n \log \frac{1}{\delta}}{\epsilon^2})$.

See you on Thurday.