Multi-Objective Weighted Sampling

Edith Cohen

[1]Google Research, CA USA

[2]School of Computer Science
Tel Aviv University, Israel

November 12, 2015
HotWeb '15

Model

Data Presentation

Data *elements* (x, w_x) have a *key* x and a numeric *value* $w_x > 0$

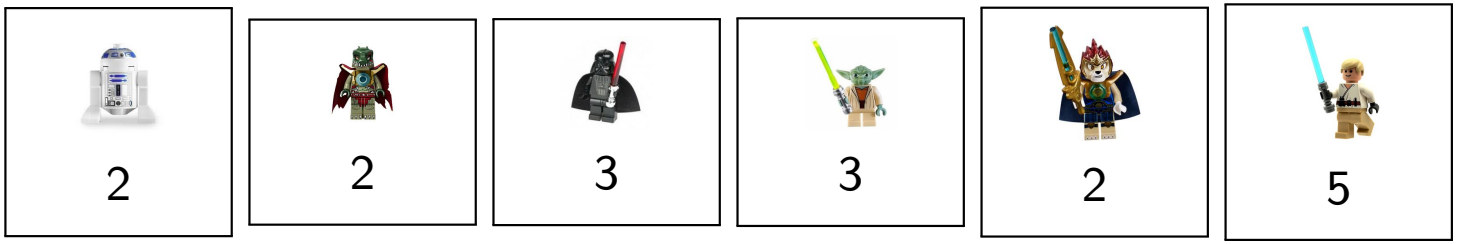
- Elements have unique keys. (Results extend when multiple elements have the same key x, and we interpret w_x as the max weight over elements.)

Model

Data Presentation

Data *elements* (x, w_x) have a *key* x and a numeric *value* $w_x > 0$

- Elements have unique keys. (Results extend when multiple elements have the same key x, and we interpret w_x as the max weight over elements.)

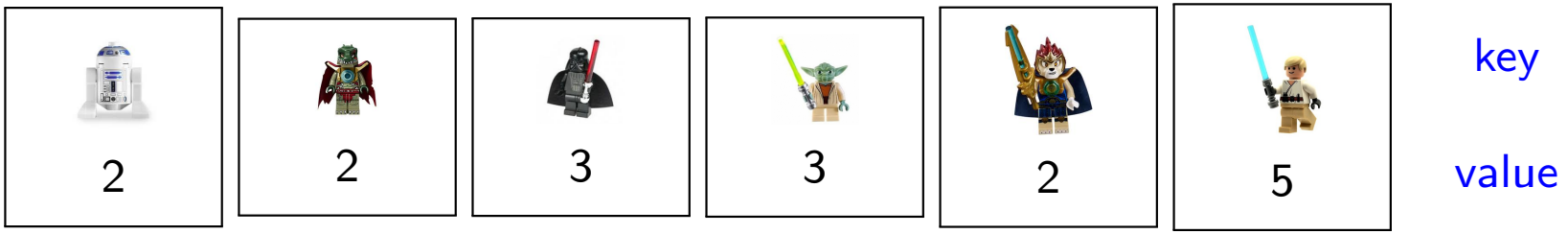


Model

Data Presentation

Data *elements* (x, w_x) have a *key* x and a numeric *value* $w_x > 0$

- Elements have unique keys. (Results extend when multiple elements have the same key x, and we interpret w_x as the max weight over elements.)



Examples

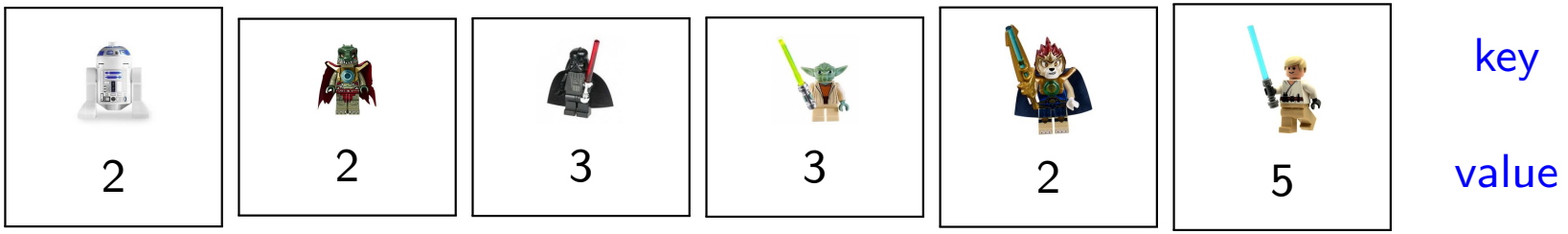
Users and activity, IP flows and sizes, Web traffic logs

Model

Data Presentation

Data *elements* (x, w_x) have a *key* x and a numeric *value* $w_x > 0$

- Elements have unique keys. (Results extend when multiple elements have the same key x, and we interpret w_x as the max weight over elements.)



Examples

Users and activity, IP flows and sizes, Web traffic logs

Queries

Queries are specified over the set of pairs (x, w_x)

Queries: Summary statistics

$$Q(f, H) = \sum_{x \in H} f(w_x)$$

- Function $f(w) \geq 0$ for $w \geq 0$ so that $f(0) = 0$
- Selected *segment* $H \subset \mathcal{X}$ (domain, subpopulation) of keys (based on demographic, location, src, dest, protocol,...)

Example $f()$:

Queries: Summary statistics

$$Q(f, H) = \sum_{x \in H} f(w_x)$$

- Function $f(w) \geq 0$ for $w \geq 0$ so that $f(0) = 0$
- Selected *segment* $H \subset \mathcal{X}$ (domain, subpopulation) of keys (based on demographic, location, src, dest, protocol,...)

Example $f()$:

- Distinct Count $f(w) = 1$ (# active keys in segment)

Queries: Summary statistics

$$Q(f, H) = \sum_{x \in H} f(w_x)$$

- Function $f(w) \geq 0$ for $w \geq 0$ so that $f(0) = 0$
- Selected *segment* $H \subset \mathcal{X}$ (domain, subpopulation) of keys (based on demographic, location, src, dest, protocol,...)

Example $f()$:

- Distinct Count $f(w) = 1$ (# active keys in segment)
- Sum $f(w) = w$ (sum of weights of keys in segment)

Queries: Summary statistics

$$Q(f, H) = \sum_{x \in H} f(w_x)$$

- Function $f(w) \geq 0$ for $w \geq 0$ so that $f(0) = 0$
- Selected *segment* $H \subset \mathcal{X}$ (domain, subpopulation) of keys (based on demographic, location, src, dest, protocol,...)

Example $f()$:

- Distinct Count $f(w) = 1$ (# active keys in segment)
- Sum $f(w) = w$ (sum of weights of keys in segment)
- Moments $f(w) = w^p$ $(p \geq 0)$ (distinct $p = 0$, sum $p = 1$)

Queries: Summary statistics

$$Q(f, H) = \sum_{x \in H} f(w_x)$$

- Function $f(w) \geq 0$ for $w \geq 0$ so that $f(0) = 0$
- Selected *segment* $H \subset \mathcal{X}$ (domain, subpopulation) of keys (based on demographic, location, src, dest, protocol,...)

Example $f()$:

- Distinct Count $f(w) = 1$ (# active keys in segment)
- Sum $f(w) = w$ (sum of weights of keys in segment)
- Moments $f(w) = w^p$ $(p \geq 0)$ (distinct $p = 0$, sum $p = 1$)
- Capping $f(w) = \mathrm{cap}_T = \min\{T, w\}$ (distinct $T = 1$, sum $T = +\infty$)

Queries: Summary statistics

$$Q(f, H) = \sum_{x \in H} f(w_x)$$

- Function $f(w) \geq 0$ for $w \geq 0$ so that $f(0) = 0$
- Selected *segment* $H \subset \mathcal{X}$ (domain, subpopulation) of keys (based on demographic, location, src, dest, protocol,...)

Example $f()$:

- Distinct Count $f(w) = 1$ (# active keys in segment)
- Sum $f(w) = w$ (sum of weights of keys in segment)
- Moments $f(w) = w^p$ $(p \geq 0)$ (distinct $p = 0$, sum $p = 1$)
- Capping $f(w) = \mathrm{cap}_T = \min\{T, w\}$ (distinct $T = 1$, sum $T = +\infty$)
- Threshold $f(w) = \mathrm{thresh}_T = I_{w \geq T}$ $(T > 0)$

Queries: Summary statistics

$$Q(f, H) = \sum_{x \in H} f(w_x)$$

- Function $f(w) \geq 0$ for $w \geq 0$ so that $f(0) = 0$
- Selected *segment* $H \subset \mathcal{X}$ (domain, subpopulation) of keys (based on demographic, location, src, dest, protocol,...)

Example $f()$:

- Distinct Count $f(w) = 1$ (# active keys in segment)
- Sum $f(w) = w$ (sum of weights of keys in segment)
- Moments $f(w) = w^p$ $(p \geq 0)$ (distinct $p = 0$, sum $p = 1$)
- Capping $f(w) = \mathrm{cap}_T = \min\{T, w\}$ (distinct $T = 1$, sum $T = +\infty$)
- Threshold $f(w) = \mathrm{thresh}_T = I_{w \geq T}$ $(T > 0)$
- Log $f(w) = \log(1 + w)$

Queries: Summary statistics

$$Q(f, H) = \sum_{x \in H} f(w_x)$$

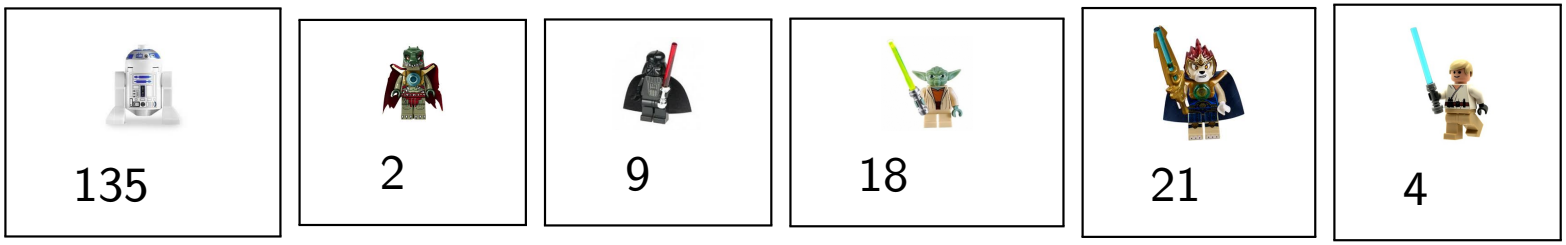
- Function $f(w) \geq 0$ for $w \geq 0$ so that $f(0) = 0$
- Selected *segment* $H \subset \mathcal{X}$ (domain, subpopulation) of keys (based on demographic, location, src, dest, protocol,...)

Example $f()$:

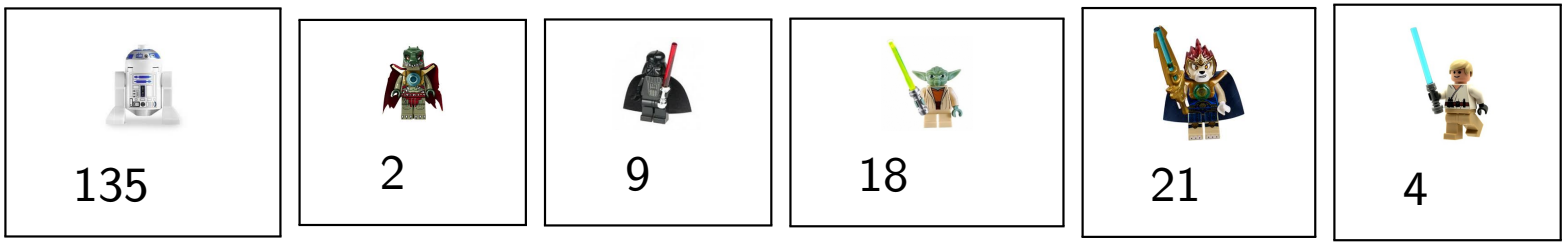
- Distinct Count $f(w) = 1$ (# active keys in segment)
- Sum $f(w) = w$ (sum of weights of keys in segment)
- Moments $f(w) = w^p$ $(p \geq 0)$ (distinct $p = 0$, sum $p = 1$)
- Capping $f(w) = \mathrm{cap}_T = \min\{T, w\}$ (distinct $T = 1$, sum $T = +\infty$)
- Threshold $f(w) = \mathrm{thresh}_T = I_{w \geq T}$ $(T > 0)$
- Log $f(w) = \log(1 + w)$

These functions are monotone non-decreasing with w.

Example queries $Q(f, H)$

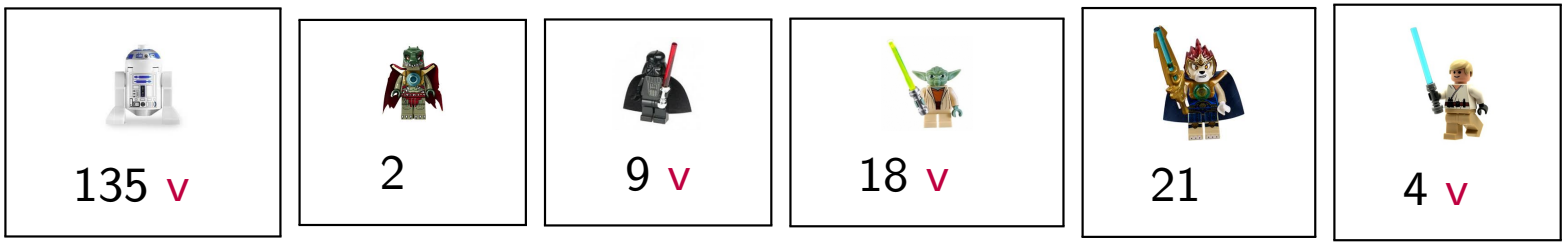


Example queries $Q(f, H)$



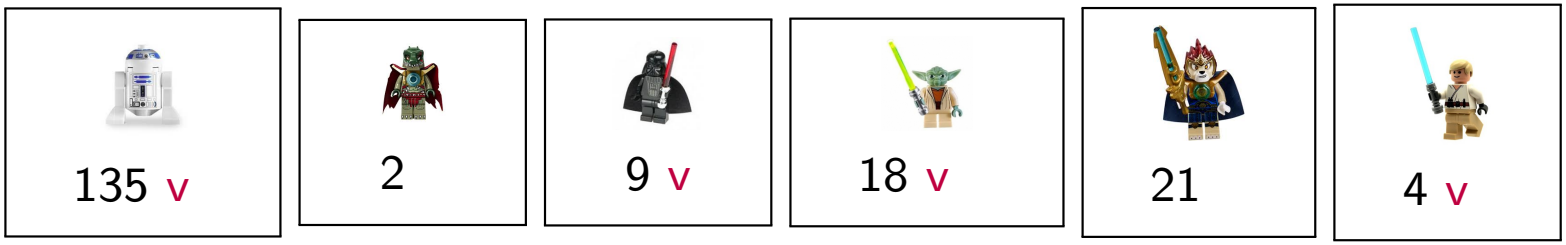
Segment H: space travelers

Example queries $Q(f, H)$



Segment H: space travelers

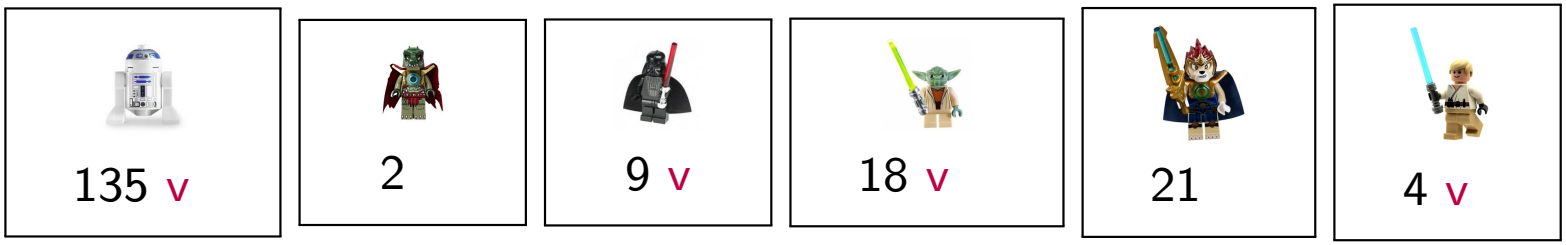
Example queries $Q(f, H)$



Segment H: space travelers

$Q(count, H) = 4$ $Q(\text{cap}_5, H) = 19$ $Q(\text{thresh}_{10}, H) = 2$

Example queries $Q(f, H)$

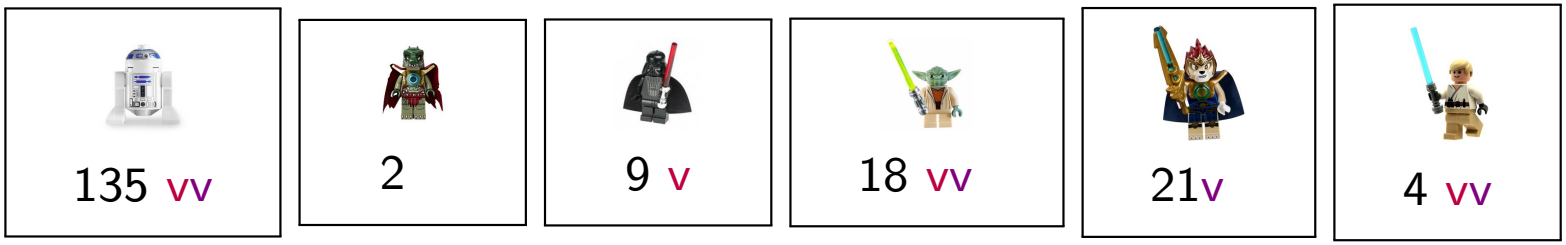


Segment H: space travelers

$Q(count, H) = 4$ $Q(\text{cap}_5, H) = 19$ $Q(\text{thresh}_{10}, H) = 2$

Segment H: Good guys

Example queries $Q(f, H)$

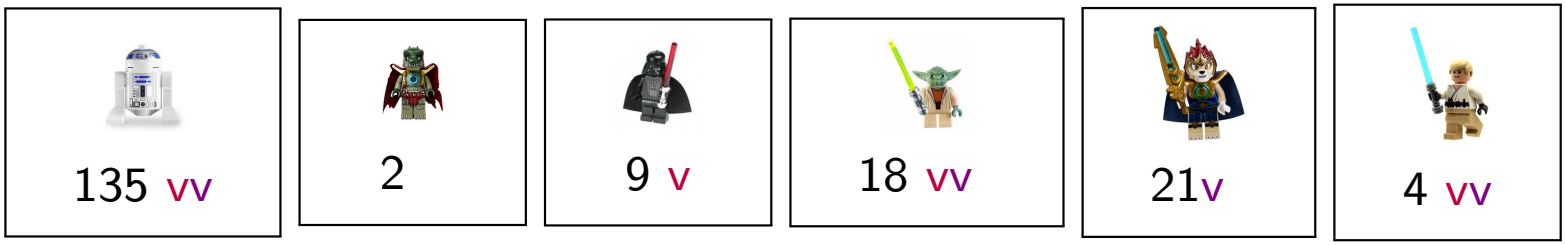


Segment H: space travelers

$Q(count, H) = 4$ $Q(\text{cap}_5, H) = 19$ $Q(\text{thresh}_{10}, H) = 2$

Segment H: Good guys

Example queries $Q(f, H)$



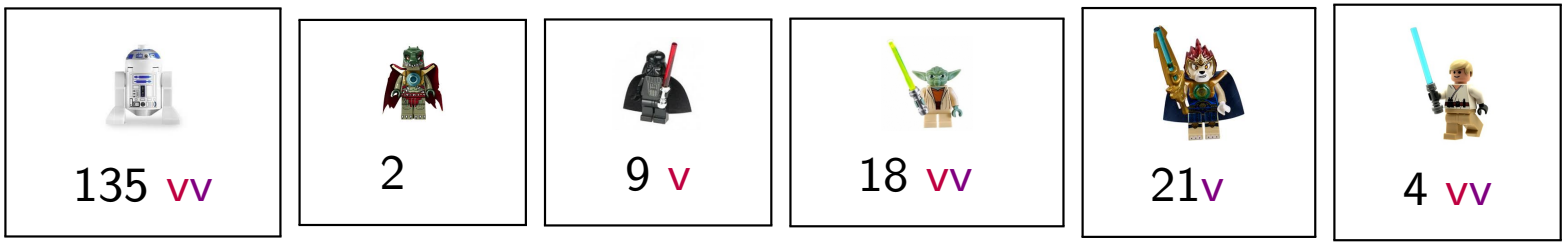
Segment H: space travelers

$Q(count, H) = 4 \qquad Q(\mathrm{cap}_5, H) = 19 \qquad Q(\mathrm{thresh}_{10}, H) = 2$

Segment H: Good guys

$Q(count, H) = 4 \qquad Q(\ln(1 + w), H) \approx 12.56$

Example queries $Q(f, H)$



Segment H: space travelers

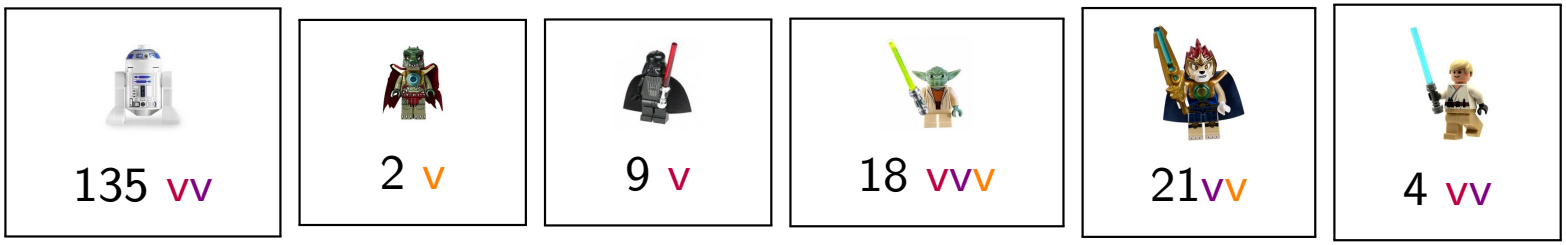
$Q(count, H) = 4 \qquad Q(\mathrm{cap}_5, H) = 19 \qquad Q(\mathrm{thresh}_{10}, H) = 2$

Segment H: Good guys

$Q(count, H) = 4 \qquad Q(\ln(1 + w), H) \approx 12.56$

Q: Segment H: Non-human life

Example queries $Q(f, H)$



Segment H: space travelers

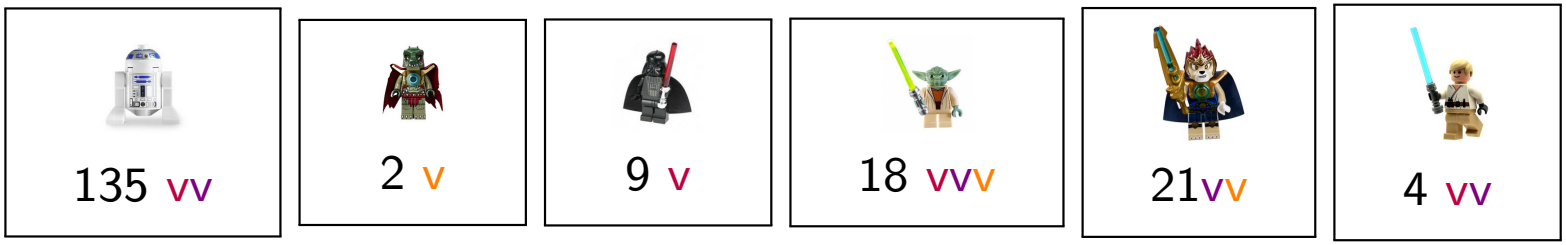
$Q(count, H) = 4$ $Q(\mathrm{cap}_5, H) = 19$ $Q(\mathrm{thresh}_{10}, H) = 2$

Segment H: Good guys

$Q(count, H) = 4$ $Q(\ln(1+w), H) \approx 12.56$

Q: Segment H: Non-human life

Example queries $Q(f, H)$



Segment H: space travelers

$Q(count, H) = 4$ $Q(\mathrm{cap}_5, H) = 19$ $Q(\mathrm{thresh}_{10}, H) = 2$

Segment H: Good guys

$Q(count, H) = 4$ $Q(\ln(1 + w), H) \approx 12.56$

Q: Segment H: Non-human life

$Q(count, H) = 3$ $Q(L_2^2, H) = 769$

Weighted Sampling

If we know the queries $Q(f, H)$ in advance, the result can be computed in a single pass.

Weighted Sampling

If we know the queries $Q(f, H)$ in advance, the result can be computed in a single pass.

Challenge: Data is massive and queries are often specified on the go, may not know H or even f in advance. We want to compute a sample from which we can quickly estimate the results.

Weighted Sampling

If we know the queries $Q(f, H)$ in advance, the result can be computed in a single pass.

Challenge: Data is massive and queries are often specified on the go, may not know H or even f in advance. We want to compute a sample from which we can quickly estimate the results.

Goals:

- Optimize tradeoffs of sample quality (statistical guarantees) and size.
- Scalable computation: A single pass on streamed or distributed elements, small state proportional to sample size

Weighted Sampling

If we know the queries $Q(f, H)$ in advance, the result can be computed in a single pass.

Challenge: Data is massive and queries are often specified on the go, may not know H or even f in advance. We want to compute a sample from which we can quickly estimate the results.

Goals:

- Optimize tradeoffs of sample quality (statistical guarantees) and size.
- Scalable computation: A single pass on streamed or distributed elements, small state proportional to sample size

When we know f, but not H in advance: Use a weighted sample computed with respect to $f(w_x)$.

Weighted Sampling

If we know the queries $Q(f, H)$ in advance, the result can be computed in a single pass.

Challenge: Data is massive and queries are often specified on the go, may not know H or even f in advance. We want to compute a sample from which we can quickly estimate the results.

Goals:

- Optimize tradeoffs of sample quality (statistical guarantees) and size.
- Scalable computation: A single pass on streamed or distributed elements, small state proportional to sample size

When we know f, but not H in advance: Use a weighted sample computed with respect to $f(w_x)$.

When we want to work with multiple f, we can use a multi-objective (MO) weighted sample, computed with respect to a set of functions F.

Weighted sampling schemes

- Data provided as key value pairs (x, w_x).
- Compute a sample S_f of size k from which we can estimate $Q(f, H)$.

Weighted sampling schemes

- Data provided as key value pairs (x, w_x).
- Compute a sample S_f of size k from which we can estimate $Q(f, H)$.

To get good size/quality tradeoffs, use (roughly) $\Pr[x \in S] \propto f(w_x)$.

Weighted sampling schemes

- Data provided as key value pairs (x, w_x).
- Compute a sample S_f of size k from which we can estimate $Q(f, H)$.

To get good size/quality tradeoffs, use (roughly) $\Pr[x \in S] \propto f(w_x)$.

- Poisson Probability Proportional to Size (PPS): Sample keys independently with $p_x = \min\{1, \frac{kf(w_x)}{\sum_x f(w_x)}\}$
- Bottom-k (order/weighted reservoir) sampling [Ros97, CK07]

Weighted sampling schemes

- Data provided as key value pairs (x, w_x).
- Compute a sample S_f of size k from which we can estimate $Q(f, H)$.

To get good size/quality tradeoffs, use (roughly) $\Pr[x \in S] \propto f(w_x)$.

- Poisson Probability Proportional to Size (PPS): Sample keys independently with $p_x = \min\{1, \frac{kf(w_x)}{\sum_x f(w_x)}\}$
- Bottom-k (order/weighted reservoir) sampling [Ros97, CK07]

foreach *key* x **do** // $Z[w]$: distribution parameterized by w
  $\mathtt{seed}(x) \sim Z[f(w_x)]$
$S \leftarrow k$ keys with smallest $\mathtt{seed}(x)$; $\tau \leftarrow (k+1)$th smallest $\mathtt{seed}(x)$

Weighted sampling schemes

- Data provided as key value pairs (x, w_x).
- Compute a sample S_f of size k from which we can estimate $Q(f, H)$.

To get good size/quality tradeoffs, use (roughly) $\Pr[x \in S] \propto f(w_x)$.

- Poisson Probability Proportional to Size (PPS): Sample keys independently with $p_x = \min\{1, \frac{kf(w_x)}{\sum_x f(w_x)}\}$
- Bottom-k (order/weighted reservoir) sampling [Ros97, CK07]

foreach *key* x **do** // $Z[w]$: distribution parameterized by w
  $\mathtt{seed}(x) \sim Z[f(w_x)]$
$S \leftarrow k$ keys with smallest $\mathtt{seed}(x)$; $\tau \leftarrow (k+1)$th smallest $\mathtt{seed}(x)$

- Sequential Poisson (priority) [Ohl98, DTL07]:
  $\mathtt{seed}(x) \sim U[0, 1/f(w_x)]$
- PPS without replacement (ppswor) [Ros72, Coh97, CK07]:
  $\mathtt{seed}(x) \sim \mathrm{Exp}[f(w_x)]$

Estimators from weighted samples

Inverse probability estimator of $Q(g, H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled

For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

Estimators from weighted samples

Inverse probability estimator of $Q(g, H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled
For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

$$\hat{Q}(g, H) = \sum_{x \in H} \hat{g}(w_x) = \sum_{x \in H \cap S} \frac{g(w_x)}{p_x} \ .$$

Estimators from weighted samples

Inverse probability estimator of $Q(g,H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled
For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

$$\hat{Q}(g,H) = \sum_{x \in H} \hat{g}(w_x) = \sum_{x \in H \cap S} \frac{g(w_x)}{p_x} \ .$$

Applies when we can compute p_x for $x \in S$

Estimators from weighted samples

Inverse probability estimator of $Q(g, H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled
For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

$$\hat{Q}(g, H) = \sum_{x \in H} \hat{g}(w_x) = \sum_{x \in H \cap S} \frac{g(w_x)}{p_x} .$$

Applies when we can compute p_x for $x \in S$

- nonnegative (since g is)
- unbiased (if $g(w_x) > 0 \implies f(w_x) > 0$)

Estimators from weighted samples

Inverse probability estimator of $Q(g, H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled
For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

$$\hat{Q}(g, H) = \sum_{x \in H} \hat{g}(w_x) = \sum_{x \in H \cap S} \frac{g(w_x)}{p_x} .$$

Applies when we can compute p_x for $x \in S$

- nonnegative (since g is)
- unbiased (if $g(w_x) > 0 \implies f(w_x) > 0$)

Poisson PPS samples: $p_x = \min\{1, \frac{kf(w_x)}{\sum_x f(w_x)}\}$
We have w_x for sampled keys $x \in S$, and the total $\sum_x f(w_x)$
$\implies$ can compute p_x and apply estimator.

Estimators from weighted samples

Inverse probability estimator of $Q(g, H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled
For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

$$\hat{Q}(g, H) = \sum_{x \in H} \hat{g}(w_x) = \sum_{x \in H \cap S} \frac{g(w_x)}{p_x} .$$

Applies when we can compute p_x for $x \in S$

- nonnegative (since g is)
- unbiased (if $g(w_x) > 0 \implies f(w_x) > 0$)

Bottom-k samples: p_x is not available so instead we use

$$p_{x|\tau} \equiv \Pr[\mathtt{seed}(x) < \tau] = \Pr[Z[f(w_x)] < \tau]$$

Estimators from weighted samples

Inverse probability estimator of $Q(g,H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled
For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

$$\hat{Q}(g,H) = \sum_{x \in H} \hat{g}(w_x) = \sum_{x \in H \cap S} \frac{g(w_x)}{p_x} .$$

Applies when we can compute p_x for $x \in S$

- nonnegative (since g is)
- unbiased (if $g(w_x) > 0 \implies f(w_x) > 0$)

Bottom-k samples: p_x is not available so instead we use

$$p_{x|\tau} \equiv \Pr[\mathtt{seed}(x) < \tau] = \Pr[Z[f(w_x)] < \tau]$$

The inclusion probability of x *conditioned* on randomization of all other keys:τ is the kth smallest $\mathtt{seed}(y)$ for $y \neq x$; $x \in S \iff \mathtt{seed}(x) < \tau$

- For ppswor $Z[y] \equiv \mathrm{Exp}[y]$: $p_{x|\tau} = 1 - e^{-f(w_x)\tau}$
- For priority $Z[y] \equiv U[0, 1/y]$: $p_{x|\tau} = \min\{f(w_x)\tau, 1\}$

Estimators from weighted samples

Inverse probability estimator of $Q(g, H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled
For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

$$\hat{Q}(g, H) = \sum_{x \in H} \hat{g}(w_x) = \sum_{x \in H \cap S} \frac{g(w_x)}{p_x} .$$

Applies when we can compute p_x for $x \in S$

- nonnegative (since g is)
- unbiased (if $g(w_x) > 0 \implies f(w_x) > 0$)

Bottom-k samples: p_x is not available so instead we use

$$p_{x|\tau} \equiv \Pr[\mathtt{seed}(x) < \tau] = \Pr[Z[f(w_x)] < \tau]$$

Estimators from weighted samples

Inverse probability estimator of $Q(g, H)$ from the sample S [HT52]

$p_x = \Pr[x \in S]$: probability that key x is sampled
For each key x, estimate $g(w_x)$ by 0 if $x \notin S$ and by $g(w_x)/p_x$ if $x \in S$.

$$\hat{Q}(g, H) = \sum_{x \in H} \hat{g}(w_x) = \sum_{x \in H \cap S} \frac{g(w_x)}{p_x} .$$

Applies when we can compute p_x for $x \in S$

- nonnegative (since g is)
- unbiased (if $g(w_x) > 0 \implies f(w_x) > 0$)

Bottom-k samples: p_x is not available so instead we use

$$p_{x|\tau} \equiv \Pr[\mathsf{seed}(x) < \tau] = \Pr[Z[f(w_x)] < \tau]$$

$$\hat{Q}(g, H) = \sum_{x \in H \cap S} \hat{g}(w_x \mid \tau) , \text{ where } \hat{g}(w_x \mid \tau) = \frac{g(w_x)}{p_{x|\tau}} .$$

How good is this estimate?

Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2

Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1

Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1
u_x	0.52	0.24	0.76	0.90	0.14	0.32	0.44	0.07	0.82

Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1
u_x	0.52	0.24	0.76	0.90	0.14	0.32	0.44	0.07	0.82

For $k = 3$, the sample is  ,  , 

Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1
u_x	0.52	0.24	0.76	0.90	0.14	0.32	0.44	0.07	0.82

For $k = 3$, the sample is  ,  , 

$\tau = 0.32$ (4^{th} smallest seed) $\Longrightarrow$

Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1
u_x	0.52	0.24	0.76	0.90	0.14	0.32	0.44	0.07	0.82

For $k = 3$, the sample is  ,  , 
$\tau = 0.32$ (4^{th} smallest seed) $\Longrightarrow$
inclusion probabilities are $p_{x|\tau} = 0.32$ for all.

...Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2

...Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
$\mathrm{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2

...Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
$\text{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2
$\frac{u_x}{\text{cap}_5(w_x)}$	0.104	0.120	0.152	0.18	0.064	0.080	0.088	0.0175	0.41

...Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
$\text{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2
$\frac{u_x}{\text{cap}_5(w_x)}$	0.104	0.120	0.152	0.18	0.064	0.080	0.088	0.0175	0.41

For $k = 3$, the sample is  ,  , 

...Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
$\text{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2
$\frac{u_x}{\text{cap}_5(w_x)}$	0.104	0.120	0.152	0.18	0.064	0.080	0.088	0.0175	0.41

For $k = 3$, the sample is  ,  , 

$\tau = 0.088$ (4^{th} smallest seed) $\Longrightarrow$

...Priority (sequential Poisson) sampling example

x									
w_x	135	2	9	18	21	4	11	4	2
$\text{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2
$\frac{u_x}{\text{cap}_5(w_x)}$	0.104	0.120	0.152	0.18	0.064	0.080	0.088	0.0175	0.41

For $k = 3$, the sample is  ,  , 
$\tau = 0.088$ (4^{th} smallest seed) $\Longrightarrow$
inclusion probabilities are

 : $p_{x|\tau} = 0.352$

 ,  : $p_{x|\tau} = 0.44$

Estimate quality when $g() = f()$

Let $q \equiv q(f, H)$ be the fraction of the statistics f due to segment H:

$$q = \frac{Q(f, H)}{Q(f, \mathcal{X})} = \frac{\sum_{x \in H} f(w_x)}{\sum_x f(w_x)} \ .$$

Estimate quality when $g() = f()$

Let $q \equiv q(f, H)$ be the fraction of the statistics f due to segment H:

$$q = \frac{Q(f, H)}{Q(f, \mathcal{X})} = \frac{\sum_{x \in H} f(w_x)}{\sum_x f(w_x)} \ .$$

bound on the Coefficient of Variation (CV) (relative standard deviation)

$$\frac{\sqrt{var[\hat{Q}(f, H)]}}{Q(f, H)} \leq \frac{1}{\sqrt{qk}}$$

Estimate quality when $g() = f()$

Let $q \equiv q(f, H)$ be the fraction of the statistics f due to segment H:

$$q = \frac{Q(f, H)}{Q(f, \mathcal{X})} = \frac{\sum_{x \in H} f(w_x)}{\sum_x f(w_x)} \ .$$

bound on the Coefficient of Variation (CV) (relative standard deviation)

$$\frac{\sqrt{var[\hat{Q}(f, H)]}}{Q(f, H)} \leq \frac{1}{\sqrt{qk}}$$

+concentration: sample size $k = c\epsilon^{-2}/q$ then prob. of rel. error $> \epsilon$ decreases exponentially in c.

Interpreting the CV bound for $g() = f()$

CV (relative standard deviation, NRMSE) bound

$$\frac{\sqrt{var[\hat{Q}(f,H)]}}{Q(f,H)} \leq \frac{1}{\sqrt{qk}}$$

Interpreting the CV bound for $g() = f()$

CV (relative standard deviation, NRMSE) bound

$$\frac{\sqrt{var[\hat{Q}(f,H)]}}{Q(f,H)} \leq \frac{1}{\sqrt{qk}}$$

$\Longrightarrow$ If we want CV $\leq \epsilon$ on segments H that have $q(f,H) \geq q$ fraction of the total f statistics, we need a sample of size $k = \epsilon^{-2}/q$

Interpreting the CV bound for $g() = f()$

CV (relative standard deviation, NRMSE) bound

$$\frac{\sqrt{var[\hat{Q}(f,H)]}}{Q(f,H)} \leq \frac{1}{\sqrt{qk}}$$

$\Longrightarrow$ If we want CV $\leq \epsilon$ on segments H that have $q(f,H) \geq q$ fraction of the total f statistics, we need a sample of size $k = \epsilon^{-2}/q$

!! This is the optimal size/quality tradeoff for sampling (on average over segments with proportion q)

For CV $\epsilon \leq 10\%$ and $q \geq 0.1\%$ $\Longrightarrow$ Sample size $k = 10^5$.

Interpreting the CV bound for $g() = f()$

CV (relative standard deviation, NRMSE) bound

$$\frac{\sqrt{var[\hat{Q}(f,H)]}}{Q(f,H)} \leq \frac{1}{\sqrt{qk}}$$

$\Longrightarrow$ If we want CV $\leq \epsilon$ on segments H that have $q(f,H) \geq q$ fraction of the total f statistics, we need a sample of size $k = \epsilon^{-2}/q$

!! This is the optimal size/quality tradeoff for sampling (on average over segments with proportion q)

For CV $\epsilon \leq 10\%$ and $q \geq 0.1\%$ $\Longrightarrow$ Sample size $k = 10^5$.

... usually $k \ll$ total number of active keys.

Estimate quality when $g() \neq f()$

We can estimate unbiasedly *any* statistics $Q(g, H)$ from a weighted sample taken with respect to f.
But what can we say about estimate quality ?

Estimate quality when $g() \neq f()$

We can estimate unbiasedly *any* statistics $Q(g, H)$ from a weighted sample taken with respect to f.
But what can we say about estimate quality ?

Disparity between g, f:

$$\rho(g, f) = \max_{w>0} \frac{g(w)}{f(w)} \max_{w>0} \frac{f(w)}{g(w)} \ .$$

Estimate quality when $g() \neq f()$

We can estimate unbiasedly *any* statistics $Q(g, H)$ from a weighted sample taken with respect to f.
But what can we say about estimate quality ?

Disparity between g, f:

$$\rho(g, f) = \max_{w>0} \frac{g(w)}{f(w)} \max_{w>0} \frac{f(w)}{g(w)} \ .$$

- Disparity is always $\rho(g, f) \geq 1$.
- We have $\rho(g, f) = 1 \iff g = cf$ for some c.

Estimate quality when $g() \neq f()$

We can estimate unbiasedly *any* statistics $Q(g, H)$ from a weighted sample taken with respect to f.
But what can we say about estimate quality ?

Disparity between g, f:

$$\rho(g, f) = \max_{w>0} \frac{g(w)}{f(w)} \max_{w>0} \frac{f(w)}{g(w)} \ .$$

- Disparity is always $\rho(g, f) \geq 1$.
- We have $\rho(g, f) = 1 \iff g = cf$ for some c.

CV of $\hat{Q}(g, H)$ is at most $(\frac{\rho}{qk})^{0.5}$.

Multi-Objective (MO) Samples

A weighted sample of size $k = \epsilon^{-2}$ with respect to f gives estimates of $Q(g, H)$ with CV $\leq \epsilon\sqrt{\rho/q}$.
$\Longrightarrow$ guarantees on quality for $Q(g, H)$ degrades with disparity $\rho(f, g)$.

What if we want CV $\leq \epsilon/\sqrt{q}$ for several $f \in F$?

Multi-Objective (MO) Samples

A weighted sample of size $k = \epsilon^{-2}$ with respect to f gives estimates of $Q(g, H)$ with CV $\leq \epsilon\sqrt{\rho/q}$.
$\Longrightarrow$ guarantees on quality for $Q(g, H)$ degrades with disparity $\rho(f, g)$.

What if we want CV $\leq \epsilon/\sqrt{q}$ for several $f \in F$?
Naive solution: Use $|F|$ independent samples S_f for $f \in F$. Size is $|F|\epsilon^{-2}$.

Multi-Objective (MO) Samples

A weighted sample of size $k = \epsilon^{-2}$ with respect to f gives estimates of $Q(g, H)$ with CV $\leq \epsilon\sqrt{\rho/q}$.
$\Longrightarrow$ guarantees on quality for $Q(g, H)$ degrades with disparity $\rho(f, g)$.

What if we want CV $\leq \epsilon/\sqrt{q}$ for several $f \in F$?
Naive solution: Use $|F|$ independent samples S_f for $f \in F$. Size is $|F|\epsilon^{-2}$.
Can we do better ?

Multi-Objective (MO) Samples

A weighted sample of size $k = \epsilon^{-2}$ with respect to f gives estimates of $Q(g, H)$ with CV $\leq \epsilon\sqrt{\rho/q}$.
$\Longrightarrow$ guarantees on quality for $Q(g, H)$ degrades with disparity $\rho(f, g)$.

What if we want CV $\leq \epsilon/\sqrt{q}$ for several $f \in F$?
Naive solution: Use $|F|$ independent samples S_f for $f \in F$. Size is $|F|\epsilon^{-2}$.
Can we do better ?

Multi-objective samples [CKS09]

Approach

Multi-Objective (MO) Samples

A weighted sample of size $k = \epsilon^{-2}$ with respect to f gives estimates of $Q(g, H)$ with CV $\leq \epsilon\sqrt{\rho/q}$.
$\Longrightarrow$ guarantees on quality for $Q(g, H)$ degrades with disparity $\rho(f, g)$.

What if we want CV $\leq \epsilon/\sqrt{q}$ for several $f \in F$?
Naive solution: Use $|F|$ independent samples S_f for $f \in F$. Size is $|F|\epsilon^{-2}$.
Can we do better ?

Multi-objective samples [CKS09]

Approach

- Make the samples for different $f \in F$ as similar as possible.
  Sample Coordination [BEJ72, Coh97]: Similar samples S_f for similar f.

Multi-Objective (MO) Samples

A weighted sample of size $k = \epsilon^{-2}$ with respect to f gives estimates of $Q(g, H)$ with CV $\leq \epsilon\sqrt{\rho/q}$.
$\Longrightarrow$ guarantees on quality for $Q(g, H)$ degrades with disparity $\rho(f, g)$.

What if we want CV $\leq \epsilon/\sqrt{q}$ for several $f \in F$?
Naive solution: Use $|F|$ independent samples S_f for $f \in F$. Size is $|F|\epsilon^{-2}$.
Can we do better ?

Multi-objective samples [CKS09]

Approach

- Make the samples for different $f \in F$ as similar as possible.
  Sample Coordination [BEJ72, Coh97]: Similar samples S_f for similar f.
- Work with a single sample, use estimators that use the inclusion probabilities in at least one sample.

Multi-Objective (MO) Samples

Multi-objective sample S_F [CKS09]

Multi-Objective (MO) Samples

Multi-objective sample S_F [CKS09]

- $S_F = \bigcup_{f \in F} S_f$ is the union of *coordinated* bottom-k (or pps) samples for $f \in F$
  E.g. with priority sampling, draw $u_x \sim U[0,1]$ once, and for S_f use $\mathsf{seed}(x) = u_x / f(w_x)$.

Multi-Objective (MO) Samples

Multi-objective sample S_F [CKS09]

- $S_F = \bigcup_{f \in F} S_f$ is the union of *coordinated* bottom-k (or pps) samples for $f \in F$
  E.g. with priority sampling, draw $u_x \sim U[0,1]$ once, and for S_f use $\mathsf{seed}(x) = u_x / f(w_x)$.
- For estimation, use $p_x = \Pr[x \in S_F]$ (inclusion in at least one dedicated S_f)

Multi-Objective (MO) Samples

Multi-objective sample S_F [CKS09]

- $S_F = \bigcup_{f \in F} S_f$ is the union of *coordinated* bottom-k (or pps) samples for $f \in F$
  E.g. with priority sampling, draw $u_x \sim U[0,1]$ once, and for S_f use $\mathsf{seed}(x) = u_x / f(w_x)$.
- For estimation, use $p_x = \Pr[x \in S_F]$ (inclusion in at least one dedicated S_f)

- Estimates have CV $\leq \epsilon/\sqrt{q}$ for $Q(f, H)$ for all $f \in F$.
- Size *typically* $\ll |F|\epsilon^{-2}$ (but is as small as possible).

Multi-objective Priority (sequential Poisson) sampling

x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1
$\text{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2
thresh_{10}	1	0	0	1	1	0	1	0	0

Multi-objective Priority (sequential Poisson) sampling

x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1
$\text{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2
thresh_{10}	1	0	0	1	1	0	1	0	0
u_x	0.52	0.24	0.76	0.90	0.14	0.32	0.44	0.07	0.82

Multi-objective Priority (sequential Poisson) sampling

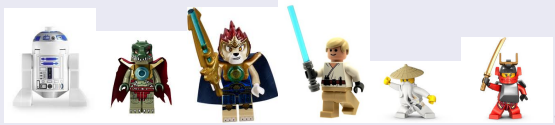
x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1
$\text{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2
thresh_{10}	1	0	0	1	1	0	1	0	0
u_x	0.52	0.24	0.76	0.90	0.14	0.32	0.44	0.07	0.82
$\frac{u_x}{\text{thresh}_{10}(w_x)}$	0.52	∞	∞	0.90	0.14	∞	0.44	∞	∞
$\frac{u_x}{\text{cap}_5(w_x)}$	0.104	0.120	0.152	0.18	0.064	0.080	0.088	0.0175	0.41

Multi-objective Priority (sequential Poisson) sampling

x									
w_x	135	2	9	18	21	4	11	4	2
Count	1	1	1	1	1	1	1	1	1
$\text{cap}_5(w_x)$	5	2	5	5	5	4	5	4	2
thresh_{10}	1	0	0	1	1	0	1	0	0
u_x	0.52	0.24	0.76	0.90	0.14	0.32	0.44	0.07	0.82
$\frac{u_x}{\text{thresh}_{10}(w_x)}$	0.52	∞	∞	0.90	0.14	∞	0.44	∞	∞
$\frac{u_x}{\text{cap}_5(w_x)}$	0.104	0.120	0.152	0.18	0.064	0.080	0.088	0.0175	0.41

For $k = 3$, the MO sample for $F = \{\text{count}, \text{thresh}_{10}, \text{cap}_5\}$ is:

MO sample: Statistical guarantees

To use the MO sample for estimating statistics, we only need to keep the sampled keys  and inclusion probabilities

- When estimating $Q(f, H)$ for any $f \in F$, the CV is at most $\frac{1}{\sqrt{qk}}$.

MO sample: Statistical guarantees

To use the MO sample for estimating statistics, we only need to keep the sampled keys  and inclusion probabilities

- When estimating $Q(f, H)$ for any $f \in F$, the CV is at most $\frac{1}{\sqrt{qk}}$.

What can we say on estimate quality f $Q(g, H)$ for $g \notin F$?

MO sample: Statistical guarantees

To use the MO sample for estimating statistics, we only need to keep the sampled keys  and inclusion probabilities

- When estimating $Q(f, H)$ for any $f \in F$, the CV is at most $\frac{1}{\sqrt{qk}}$.

What can we say on estimate quality f $Q(g, H)$ for $g \notin F$?

MO sample is at least as good as any of the dedicated samples it includes $\Longrightarrow$ CV is at most $\sqrt{\frac{\min_{f \in f} \rho(f,g)}{qk}}$.

MO sample: Statistical guarantees

To use the MO sample for estimating statistics, we only need to keep the sampled keys  and inclusion probabilities

- When estimating $Q(f, H)$ for any $f \in F$, the CV is at most $\frac{1}{\sqrt{qk}}$.

What can we say on estimate quality f $Q(g, H)$ for $g \notin F$?

MO sample is at least as good as any of the dedicated samples it includes $\Longrightarrow$ CV is at most $\sqrt{\frac{\min_{f \in f} \rho(f,g)}{qk}}$.

Theorem: When g is a nonnegative combination of functions from F, estimates of $Q(g, H)$ from S_F have CV at most $\frac{1}{\sqrt{qk}}$.

Computing the MO sample

The MO samples are composable: The sample of a union of sets of elements can be obtained from the samples of the sets.

Computing the MO sample

The MO samples are composable: The sample of a union of sets of elements can be obtained from the samples of the sets.

This means that MO sampling can be performed efficiently in streaming and distributed settings: Single pass with state proportional to sample size.

Computing the MO sample

The MO samples are composable: The sample of a union of sets of elements can be obtained from the samples of the sets.

This means that MO sampling can be performed efficiently in streaming and distributed settings: Single pass with state proportional to sample size.

But when computing the sample for arbitrary F, we still need to test each key for possible membership in the sample, which is at least $O(|F|)$.

Challenge

What can we say on both sample size and computation for some natural classes of functions

- All monotone non-decreasing f ?
- All Capping functions ?

Challenge

What can we say on both sample size and computation for some natural classes of functions

- All monotone non-decreasing f ?
- All Capping functions ?

$|F|$ is very large, we do not want the computation to depend on F.

Universal sample for all monotone functions

MO sampling the set M of all monotone non-decreasing functions $f(w_x)$

Universal sample for all monotone functions

MO sampling the set M of all monotone non-decreasing functions $f(w_x)$

M includes all moment, capping, threshold functions and more ...

Universal sample for all monotone functions

MO sampling the set M of all monotone non-decreasing functions $f(w_x)$

M includes all moment, capping, threshold functions and more ...

Theorem [Coh15b]

Universal sample for all monotone functions

MO sampling the set M of all monotone non-decreasing functions $f(w_x)$

M includes all moment, capping, threshold functions and more ...

Theorem [Coh15b]

- Size: $\mathsf{E}[|S_M|] \leq \epsilon^{-2} \ln n$, where n is the number of keys.

Universal sample for all monotone functions

MO sampling the set M of all monotone non-decreasing functions $f(w_x)$

M includes all moment, capping, threshold functions and more ...

Theorem [Coh15b]

- Size: $\mathsf{E}[|S_M|] \leq \epsilon^{-2} \ln n$, where n is the number of keys.
- Computation: S_M and inclusion probabilities used for estimation can be computed using $O(n \log \epsilon^{-1})$ operations.

Universal sample for all monotone functions

MO sampling the set M of all monotone non-decreasing functions $f(w_x)$

M includes all moment, capping, threshold functions and more ...

Theorem [Coh15b]

- Size: $\mathsf{E}[|S_M|] \leq \epsilon^{-2} \ln n$, where n is the number of keys.
- Computation: S_M and inclusion probabilities used for estimation can be computed using $O(n \log \epsilon^{-1})$ operations.
- Tight lower bound: When keys have distinct weights, any sample providing these statistical guarantees has size $\Omega(\epsilon^{-2} \ln n)$.
  Enough to look at thresh functions ($\mathrm{thresh}_T(x) = 1$ if $x \geq T$ and 0 otherwise)

Universal sample for all monotone functions

MO sampling the set M of all monotone non-decreasing functions $f(w_x)$

M includes all moment, capping, threshold functions and more ...

Theorem [Coh15b]

- Size: $\mathsf{E}[|S_M|] \leq \epsilon^{-2} \ln n$, where n is the number of keys.
- Computation: S_M and inclusion probabilities used for estimation can be computed using $O(n \log \epsilon^{-1})$ operations.
- Tight lower bound: When keys have distinct weights, any sample providing these statistical guarantees has size $\Omega(\epsilon^{-2} \ln n)$.
  Enough to look at thresh functions ($\mathrm{thresh}_T(x) = 1$ if $x \geq T$ and 0 otherwise)

Sampling scheme builds on a surprising relation to computing All-Distances sketches [Coh97, Coh15a])

Universal monotone sample: Characterization

Key idea in sampling scheme design:
A simple characterization of the sample S_M, which enables us to compute it very efficiently.

If we sort all keys by decreasing weights, a key x is included in S_M if and only if its random hash u_x is among the k smallest hash values of keys in the prefix.

Universal monotone sample: Characterization

Key idea in sampling scheme design:
A simple characterization of the sample S_M, which enables us to compute it very efficiently.

If we sort all keys by decreasing weights, a key x is included in S_M if and only if its random hash u_x is among the k smallest hash values of keys in the prefix.

Can also compute the inclusion probabilities efficiently.

Universal function for all capping functions

$\mathrm{cap}_T(w) = \min\{T, w\}$

- $C \subset M$ thus $S_C \subset S_M$
- Efficient sampling scheme for S_C: Identify all keys "eligible" for inclusion in S_M and trim down.
- Typically $|S_C| \ll |S_M|$. Therefore, if we are only interested in nonnegative combinations of capping functions, we should work with S_C.

Conclusion

- Weighted sampling is a powerful way to summarize massive data.
- Unbiased estimates for all statistics, but strong statistical guarantees on quality only for the particular "weights" $f(w_x)$ (or very similar weights).
- With multi-objective weighted samples we can provide desired statistical guarantees for different "objectives" while minimizing the required summary size.

Conclusion

- Weighted sampling is a powerful way to summarize massive data.
- Unbiased estimates for all statistics, but strong statistical guarantees on quality only for the particular "weights" $f(w_x)$ (or very similar weights).
- With multi-objective weighted samples we can provide desired statistical guarantees for different "objectives" while minimizing the required summary size.

New results:

Conclusion

- Weighted sampling is a powerful way to summarize massive data.
- Unbiased estimates for all statistics, but strong statistical guarantees on quality only for the particular "weights" $f(w_x)$ (or very similar weights).
- With multi-objective weighted samples we can provide desired statistical guarantees for different "objectives" while minimizing the required summary size.

New results:

- Basic closure property of MO samples S_F: Statistical guarantees extend to for nonnegative linear combinations of function in F.

Conclusion

- Weighted sampling is a powerful way to summarize massive data.
- Unbiased estimates for all statistics, but strong statistical guarantees on quality only for the particular "weights" $f(w_x)$ (or very similar weights).
- With multi-objective weighted samples we can provide desired statistical guarantees for different "objectives" while minimizing the required summary size.

New results:

- Basic closure property of MO samples S_F: Statistical guarantees extend to for nonnegative linear combinations of function in F.
- Efficient universal sampling scheme for the the set M of all monotone non-decreasing functions.

Conclusion

- Weighted sampling is a powerful way to summarize massive data.
- Unbiased estimates for all statistics, but strong statistical guarantees on quality only for the particular "weights" $f(w_x)$ (or very similar weights).
- With multi-objective weighted samples we can provide desired statistical guarantees for different "objectives" while minimizing the required summary size.

New results:

- Basic closure property of MO samples S_F: Statistical guarantees extend to for nonnegative linear combinations of function in F.
- Efficient universal sampling scheme for the the set M of all monotone non-decreasing functions.
- Showed that S_M is larger than a dedicated sample (which provides the quality guarantees for a single f), by at most a $\ln n$ factor.
  + Tight lower bound on $|S_M|$ when weights are distinct

Conclusion

- Weighted sampling is a powerful way to summarize massive data.
- Unbiased estimates for all statistics, but strong statistical guarantees on quality only for the particular "weights" $f(w_x)$ (or very similar weights).
- With multi-objective weighted samples we can provide desired statistical guarantees for different "objectives" while minimizing the required summary size.

New results:

- Basic closure property of MO samples S_F: Statistical guarantees extend to for nonnegative linear combinations of function in F.
- Efficient universal sampling scheme for the the set M of all monotone non-decreasing functions.
- Showed that S_M is larger than a dedicated sample (which provides the quality guarantees for a single f), by at most a $\ln n$ factor.
  $+$ Tight lower bound on $|S_M|$ when weights are distinct
- Efficient sampling scheme for the universal capping sample

Thank you!!

Bibliography I

K. R. W. Brewer, L. J. Early, and S. F. Joyce.
Selecting several samples from a single population.
Australian Journal of Statistics, 14(3):231–239, 1972.

E. Cohen and H. Kaplan.
Summarizing data using bottom-k sketches.
In *ACM PODC*, 2007.

E. Cohen, H. Kaplan, and S. Sen.
Coordinated weighted sampling for estimating aggregates over multiple weight assignments.
VLDB, 2(1–2), 2009.
full: `http://arxiv.org/abs/0906.4560`.

E. Cohen.
Size-estimation framework with applications to transitive closure and reachability.
J. Comput. System Sci., 55:441–453, 1997.

E. Cohen.
All-distances sketches, revisited: HIP estimators for massive graphs analysis.
TKDE, 2015.

E. Cohen.
Multi-objective weighted sampling.
In *HotWeb*. IEEE, 2015.
full version: `http://arxiv.org/abs/1509.07445`.

Bibliography II


N. Duffield, M. Thorup, and C. Lund.
Priority sampling for estimating arbitrary subset sums.
J. Assoc. Comput. Mach., 54(6), 2007.


D. G. Horvitz and D. J. Thompson.
A generalization of sampling without replacement from a finite universe.
Journal of the American Statistical Association, 47(260):663–685, 1952.


E. Ohlsson.
Sequential poisson sampling.
J. Official Statistics, 14(2):149–162, 1998.


B. Rosén.
Asymptotic theory for successive sampling with varying probabilities without replacement, I.
The Annals of Mathematical Statistics, 43(2):373–397, 1972.


B. Rosén.
Asymptotic theory for order sampling.
J. Statistical Planning and Inference, 62(2):135–158, 1997.