Sampling Big Ideas in Sublinear Algorithms

Edith Cohen

Google Research

Tel Aviv University

Random Sampling

Basis of learning from observations

Recent centuries: Tool for efficiently surveying populations

- Graunt 1662: Estimate population of England
- Laplace ratio estimator [1786, 1802]: Estimate the population of France. Sampled "Communes" (administrative districts), counting population ratio to live births in previous year. Extrapolate from birth registrations in whole country.
- Kiaer 1895 "representative method"; March 1903 "probability sampling"
- US census 1938: Use probability sample to estimate unemployment

⋮

Recent decades: Ubiquitous tool in efficient data processing, analysis, algorithms design

Big Ideas – Sampling Toolbox

This talk: Some synthesis of **selected** ideas
Originated in stats and CS works – see writeup for references

- Composable summary structures
- PPS sampling for Linear Queries
- Sampling via Order Statistics
  - Graph sketches
  - Estimation of Linear Queries
  - Soft PPS
  - Sticky per-key randomness
    - Coordinated Samples
    - Multi-objective Samples

Sampling Unaggregated raw data

- Max aggregation
- Sum aggregation
- Functions of frequency
  - Transform to Max
  - Transform to Heavy Hitters

Sample as a Summary Structure

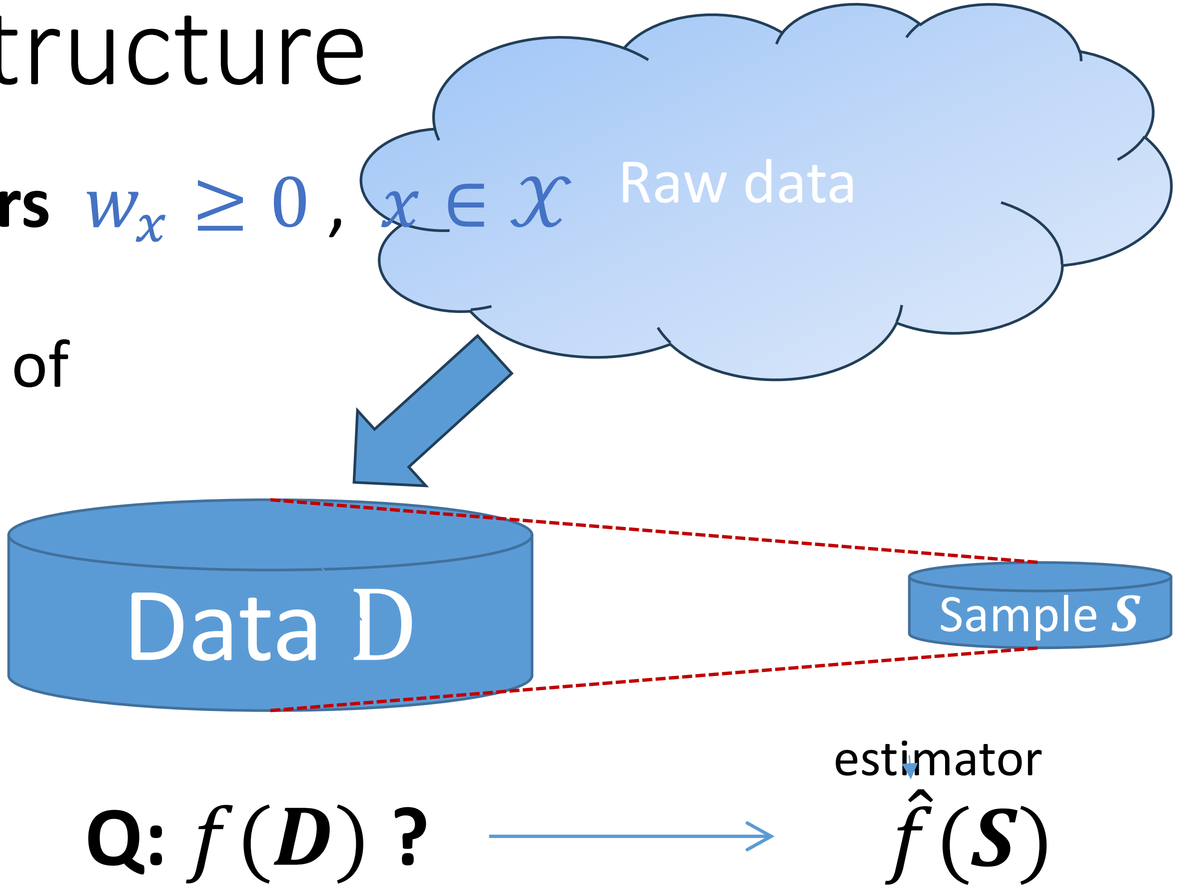
Dataset $D = \{(x, w_x)\}$ of **key-value pairs** $w_x \geq 0$, $x \in \mathcal{X}$

Sampling Scheme specifies a randomized summary structure $\mathrm{D} \mapsto S(D)$ in the form of a subset of keys and auxiliary information

Approximate Queries in Sample Space:

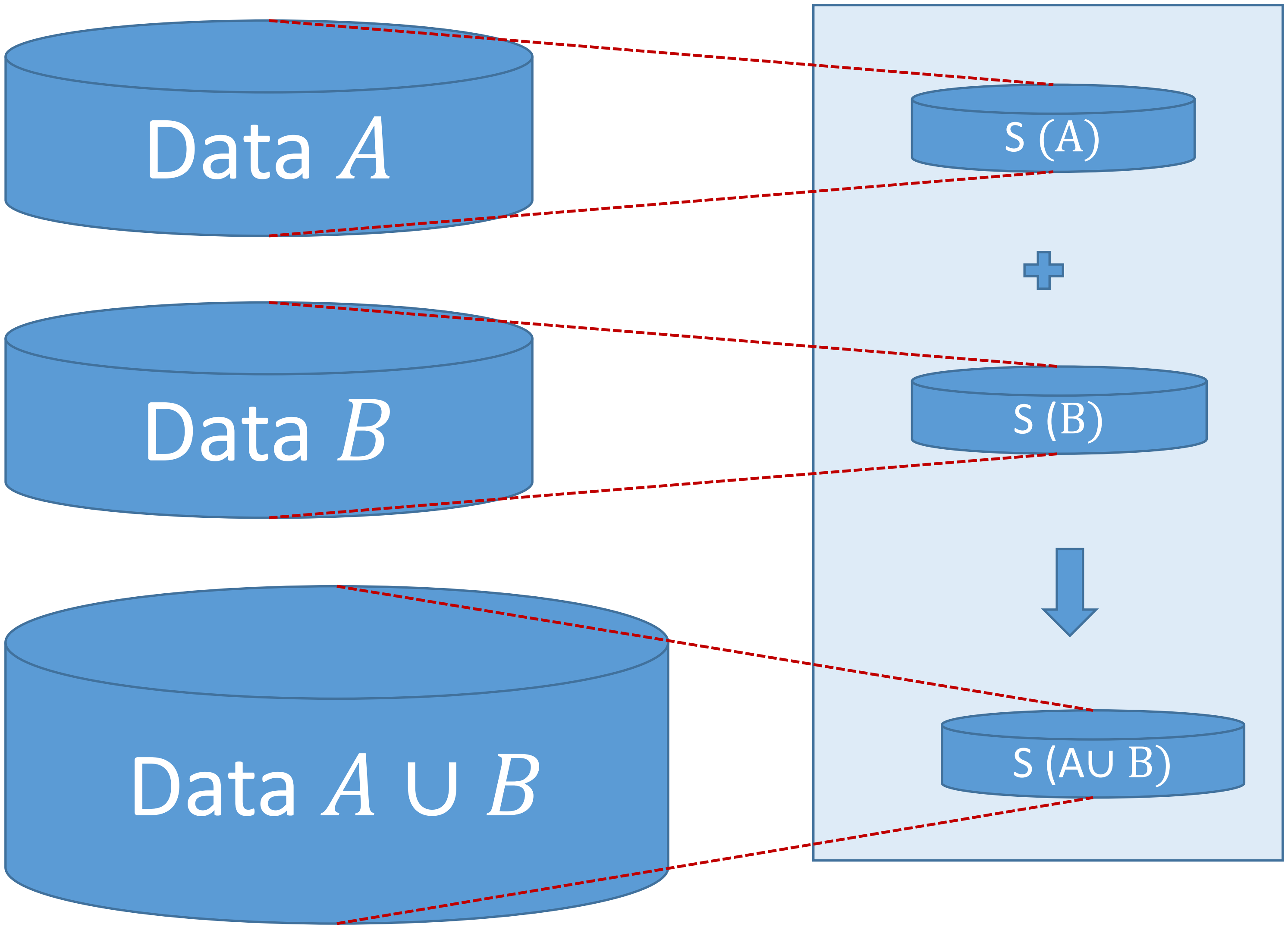
$f(D)$ from $S(D)$

$f(D_1, D_2, \ldots)$ from $S(D_1), S(D_2), \ldots$



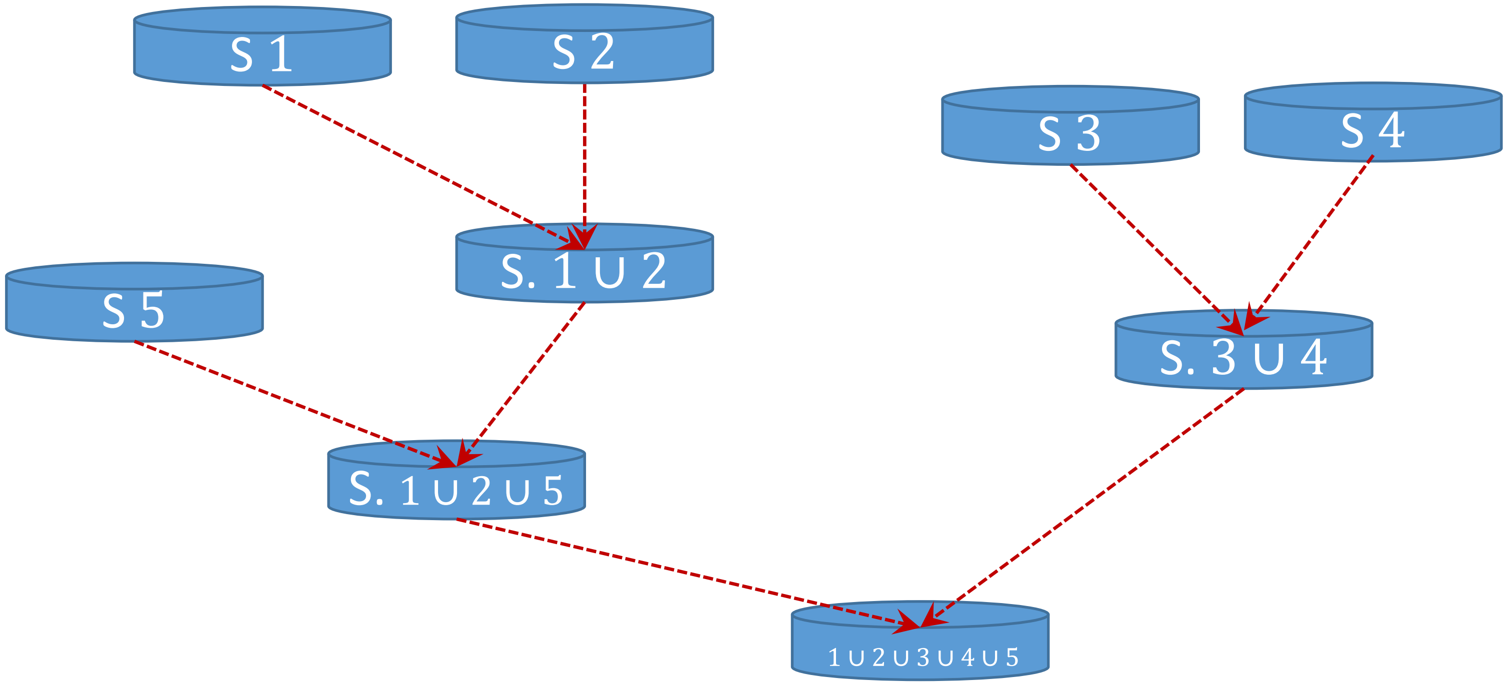
- Sampling schemes and estimators (query response algorithms) (Accuracy vs. sample size)
- Efficient Algorithms (storage, communication, compute)
  - over the raw data presentation (distributed, streaming, graph, unaggregated…)

Composable (Mergeable) Summary Structures

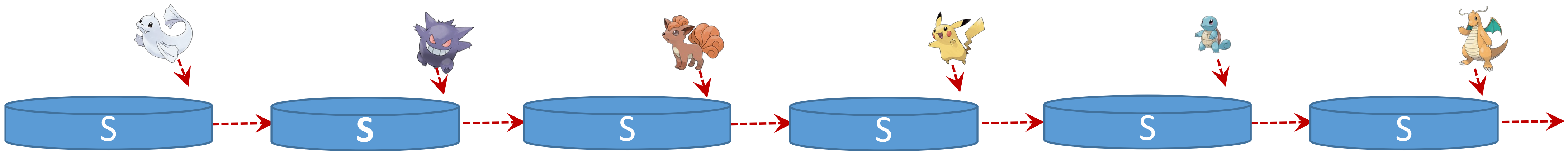


Why Composable ?

Distributed data/parallelize computation



Streaming



Task: Linear Aggregation Queries

Dataset $D = \{(x, w_x)\}\ w_x \geq 0\ ,\ x \in \mathcal{X}$

Sample $S(D)$

Query: $h: \mathcal{X} \rightarrow \{0,1\}$ **Response:** estimate $sum(h) \coloneqq \sum_x h(x) \cdot w_x$ from $S(D)$

Error bounds* for sample size k

$$q_h \coloneqq \frac{sum(h)}{sum(\mathbf{1})} \qquad \text{NRMSE} \ \frac{\sigma}{sum(h)} \leq \frac{1}{\sqrt{k} \cdot q_h}$$

*With Probability Proportional to Size (PPS) sampling

(Weighted/Importance sampling)

PPS Sampling Schemes

Sample size k

PPSWR (With Replacement):

Repeat k times:

Select key x with prob. $\frac{w_x}{sum(1)}$

PPSWOR (WithOut Replacement)

$Y \leftarrow \mathcal{X}$

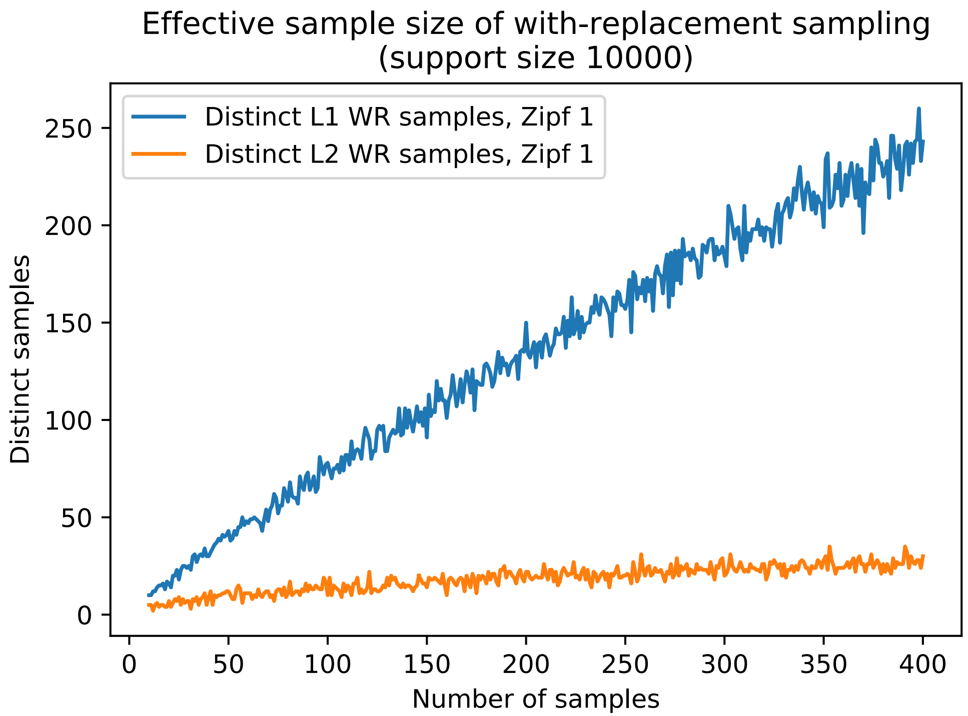
Repeat k times:

Select key $x \in Y$ with prob $\frac{w_x}{sum(1_Y)}$

Remove x from Y

Stochastic partition: random partition of $\mathcal{X}$ to k parts. PPS sample from each part

- All yield the same worst-case error bounds
- PPSWR less update-efficient
- PPSWOR better on "heavy tailed" data



Big Ideas – Outline

- Composable summary structures
- PPS sampling for Linear Queries
- Sampling via Order Statistics
  - Distributed/streaming use
  - Graph sketches
  - Estimation of Linear Queries
  - Soft PPS
  - Sticky per-key randomness
    - Coordinated Samples
    - Multi-objective Samples

Sampling Unaggregated raw data

- Max aggregation
- Sum aggregation
- Functions of frequency
  - Transform to Max
  - Transform to Heavy Hitters

Sampling via Order Statistics

Transform sampling problem → computing bottom-k order statistics

Gains: Efficiency -- PPSWOR as described is sequential and inefficient

Applicability -- e.g. sample without knowledge of $sum(1) := \sum_{x \in H} w_x$

Compute an (independent) score to each (key,value) pair

$e = (x, w_x) \mapsto score(e) \sim F[w_x]$

$$F[w] := \begin{cases} U\left[0, \frac{1}{w}\right] & ; \Rightarrow \text{sequential Poisson (Priority sampling)} \\ \text{Exp}[w] & ; \Rightarrow \text{PPSWOR.} \end{cases}$$

PPSWR ↦ bottom-1 score ($\times\, k$)

PPSWOR ↦ bottom-k scores

Bottom-k Transform: Why we get PPSWOR

$$score(i, w_i) = X_i \sim Exp[w_i]$$

First step:

What is $\Pr\left[X_1 < \min_{i>1} X_i\right]$?

(ii) $\Rightarrow \min_{i>1} X_i \sim \mathrm{Exp}[\sum_{i>1} w_i]$

(i) $\Rightarrow \Pr\left[X_1 < \min_{i>1} X_i\right] = \frac{w_1}{w_1 + \sum_{i>1} w_i} = \frac{w_1}{\sum_i w_i}$

Properties of Exponential Distribution

$$X_i \sim \mathrm{Exp}[w_i]$$

(i) $\Pr[X_1 < X_2] = \frac{w_1}{w_1 + w_2}$

(ii) $\min_i X_i \sim \mathrm{Exp}[\sum_i w_i]$

(iii) Memorylessness:

$\Pr[X > t_0 + t \mid X > t\_0] = \Pr[X > t]$

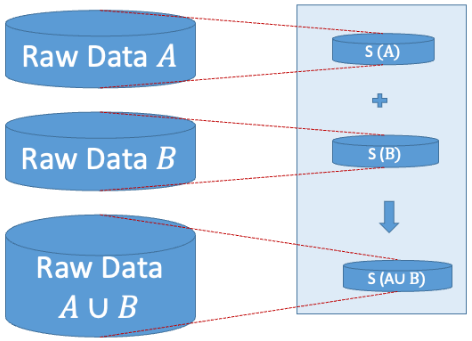
Step 2+:

(iii) $\Rightarrow$ For $t \geq 0, i > 1$ $\quad \Pr[X_i > X_1 + t \mid X_i > X_1] = \Pr[X_i > t]$

Bottom-k Transform: Efficiency benefits

$$e = (x, w_x) \mapsto score(e) \sim F[w_x]$$

Application:
Raw data presents as distributed or streaming of pairs $\{(x, w_x)\}$



- Score is ``locally'' computed for each pair
- Composability of bottom- k :
  bottom-$k(A \cup B)$ = bottom-k(bottom-$k(A)$ ∪ bottom−k(B))

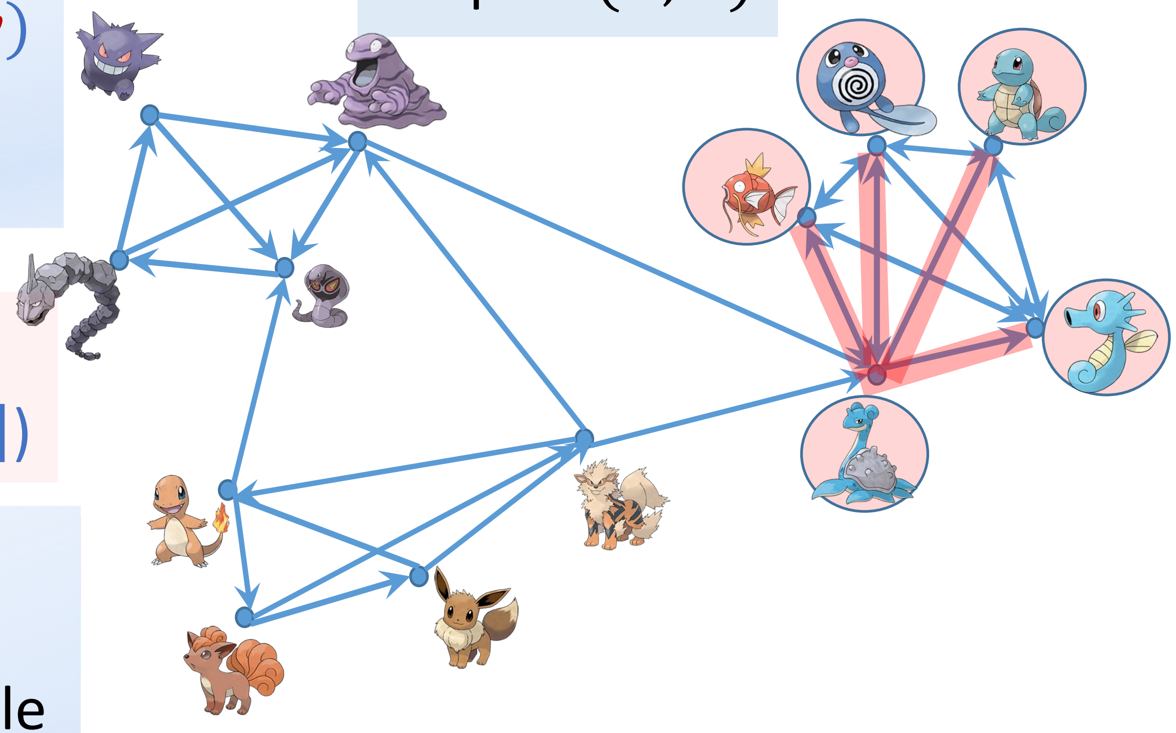
More Applications: Neighborhood samples for each node in a graph

Graphs Sketches: Node-Centric samples of neighborhoods or reachability sets

Task: Compute for *each* node v, a sample $S(v)$

- Of $N(v)$: nodes reachable from v
- Of $N(v)$: nodes within distance 5 from v

Graph $G(V,E)$



- Naïve: first compute $N(v)$: $O(|E||V|)$
- Through order transform: near-linear $\tilde{O}(k\,|E|)$

Idea:

- $score(v) \sim F$ for each $v \in V$
- Compute for each node $u \in V$ the k reachable nodes v with smallest $score(v)$

"Propagate" scores by prioritizing lower values.
Each node is visited at most k times.

- Samples are computed without knowing cardinality $|N(v)|$!
- Can be used to estimate it!

Big Ideas – Outline

- Composable summary structures
- PPS sampling for Linear Queries
- Sampling via Order Statistics
  - Distributed/streaming use
  - Graph sketches
  - Estimation of Linear Queries
  - Soft PPS
  - Sticky per-key randomness
    - Coordinated Samples
    - Multi-objective Samples

Sampling Unaggregated raw data

- Max aggregation
- Sum aggregation
- Functions of frequency
  - Transform to Max
  - Transform to Heavy Hitters

Sum Estimation as Parameter estimation

Properties of Exponential Distribution

$X_i \sim \mathrm{Exp}[w_i]$

(ii) $\min_i X_i \sim \mathrm{Exp}[\sum_i w_i]$

Estimating $sum(1)$ from order samples:

The minimum score is iid $\mathrm{Exp}[sum(1)]$

PPSWR: The minimum scores in the k reps are k iid samples $\mathrm{r}_\mathrm{i} \sim \mathrm{Exp}[sum(1)]$

The (optimal) unbiased estimator is $\frac{k-1}{\sum_{i\in[k]} r_i}$ NRMSE is $\frac{1}{\sqrt{k-1}}$

PPSWOR sample: can use memorylessness to "extract" k iid samples r_i $\sim \mathrm{Exp}[sum(1)]$ from the bottom- k scores and weights of sampled keys.

Estimating Linear Queries from Samples

$$sum(h) := \sum_x h(x) \cdot w_x$$

Inverse probability per-key estimate: $\widehat{w_x} = \begin{cases} \frac{w_x}{\Pr[x \in S]} & x \in S \\ 0 & x \notin S \end{cases}$

$$\widehat{\mathrm{Sum}(h)} = \sum_x h(x) \cdot \widehat{w_x} = \sum_{x \in S} h(x) \cdot \widehat{w_x}$$

Per-key estimates are **unbiased** (if $w_x > 0 \Rightarrow \Pr[x \in S\} > 0$)

$$E[\widehat{w_x}] = 0 \cdot \Pr[x \notin S] + \frac{w_x}{\Pr[x \in S]} \cdot \Pr[x \in S] = \mathrm{w_x}$$

linearity of expectation ⇒ sum estimate is unbiased

$$E[\widehat{\mathrm{Sum}(h)}] = \sum_x h(x) \cdot \mathrm{E}[\widehat{w_x}] = \sum_x h(x) \cdot w_x = \mathrm{Sum}(h)$$

!! With PPS schemes we get ~optimal error bounds.

Catch -- need to compute $\mathbf{Pr}[\boldsymbol{x} \in \boldsymbol{S}]$. With PPSWR samples, we need $sum(\mathbf{1})$. For bottom-k samples, $\mathbf{Pr}[\boldsymbol{x} \in \boldsymbol{S}\,]$ depends weights of keys not in sample.

Inverse Probability with bottom-k samples

Idea: Detangle -- use $\mathbf{Pr}[\boldsymbol{x} \in \boldsymbol{S}\,]$ conditioned on the scores of all other keys

$\tau_x \coloneqq \{score(e) \mid e.key \neq x\}_{(k)}$ The kth lowest score of keys other than x

Now easy to compute:

$$\Pr[\,x \in S \mid \tau_x = t\,] = \Pr[score(x) < t] = 1 - e^{-w_x \cdot t}$$

For PPSWOR

$$\widehat{w_x} = \begin{cases} \dfrac{w_x}{\Pr[\,x \in S \mid \tau_x = t\,]} & x \in S \\ 0 & x \notin S \end{cases}$$

Unbiased when conditioned on $\tau_x \implies$ unbiased (over distribution of τ_x)

For $x \in S$, $\tau_x = \tau \coloneqq \{score(e)\}_{k+1}$

To facilitate estimation, we store the lowest score with the sample

Error bounds? Conditioning increases variance

Inverse Probability with PPSWOR --

Still get ~optimal error bounds!

$$q_h := \frac{sum(h)}{sum(1)} \quad \frac{\sqrt{\mathrm{Var}[\widehat{\mathrm{Sum}}(h)]}}{sum(h)} \leq \frac{1}{\sqrt{k-1} \cdot q_h}$$

- Bound per-key variance $\mathrm{var}[\widehat{w_x}] \leq \frac{1}{k-1} w_x \cdot sum(1)$

  $$\mathrm{var}[\,\widehat{w_x} \mid \tau_x = t\,] \leq \frac{w_x}{t}$$

  $$\mathrm{var}[\,\widehat{w_x}] = \mathrm{E}_{\tau_x} \mathrm{var}[\,\widehat{w_x} \mid \tau_x = t\,]$$

  Distribution of τ_x is "dominated" by sum of k iid Exp $[sum(1)]$ (small val more likely)

- $\mathrm{cov}[\widehat{w_x}, \widehat{w_y}] \leq 0$

$$\Rightarrow var[\widehat{Sum}(h)] \leq \sum_{x \in H} var[\widehat{w_x}]$$

$$\leq \frac{1}{k-1} \cdot sum(1) \cdot \sum_{x \in H} w_x = \frac{1}{k-1} sum(h)\, sum(1)$$

Big Ideas – Outline

- Composable summary structures
- PPS sampling for Linear Queries
- Sampling via Order Statistics
  - Distributed/streaming use
  - Graph sketches
  - Estimation of Linear Queries
  - Soft PPS
  - Sticky per-key randomness
    - Coordinated Samples
    - Multi-objective Samples

Sampling Unaggregated raw data

- Max aggregation
- Sum aggregation
- Functions of frequency
  - Transform to Max
  - Transform to Heavy Hitters

Soft PPS

Scenario: We PPS sampled with respect to weights w_x
but our query is $\sum_x h(x) \cdot w'_x$ with respect to weights w'_x

Useful when

- we are limited by the sampling procedure
- weights change and we want to use the same sample

Inverse probability estimate: $\widehat{w'}_x = \begin{cases} \frac{w'_x}{\Pr[x \in S]} & x \in S \\ 0 & x \notin S \end{cases}$

Unbiased when
$w'_x > 0 \Rightarrow \Pr[x \in S] > 0$)

Error bound: $\frac{\rho}{\sqrt{k} \cdot q_h}$

$$\rho(\boldsymbol{w}', \boldsymbol{w}) := \frac{\text{sum}_{\text{w}}(1)}{\text{sum}_{\text{w}'}(1)} \cdot \max_x \frac{w'_x}{w_x}$$

Takeaway: gracefully degrades with ``distance" between $\boldsymbol{w}'$ and $\boldsymbol{w}$
can get the "exact PPS" error – but with a larger sample

Big Ideas – Outline

- Composable summary structures
- PPS sampling for Linear Queries
- Sampling via Order Statistics
  - Distributed/streaming use
  - Graph sketches
  - Estimation of Linear Queries
  - Soft PPS
  - Sticky per-key randomness
    - Coordinated Samples
    - Multi-objective Samples

Sampling Unaggregated raw data

- Max aggregation
- Sum aggregation
- Functions of frequency
  - Transform to Max
  - Transform to Heavy Hitters

Sticky per-key randomness

Permanent Random Numbers (PRN) [Brewer, Early, Joyce 1972]

Idea: "Attach" the random bits of $score(e = (x, w_x))$ to the key x (rather than the pair)

For each key x, iid $\mathrm{seed}(x) \sim F$

$$score(e) \leftarrow \frac{1}{w_x} \cdot seed(x)$$

$$\mathrm{Exp}[w] \equiv \frac{1}{w} \cdot \mathrm{Exp}[1]$$

$$U\left[0, \frac{1}{w}\right] \equiv \frac{1}{w} \cdot U[0,1]$$

Application scenarios:

- When there are multiple contexts/weights (aka instances) for the same set of keys
  Samples of instances with sticky randomness are coordinated

- Unaggregated data – key appears in multiple data elements (e.g. distinct counting)

Sample Coordination

Same set of keys, multiple sets of weights ("instances"). Sample each instance

Monday:

Tuesday:

Coordinated Samples: Benefits

- Stability of sample under dynamic weight changes

Survey sampling: Weights evolve but surveys impose burden. Want to minimize the burden and maintain a PPS sample of the evolved set.

- Representations (sketches) of instances:
  - Accuracy on inter-instance queries (e.g. similarity)
    - Locality Sensitive Hashing (LSH) property (similar weights$\Rightarrow$ similar samples)
  - Efficient to compute for many instances (e.g. Graph sketches)
- Maximize agreement when sampling from different instances (improved privacy/utility)
- Multi-objective samples – overlap of samples implies less storage/computation

Multi-objective Sample: Basic

Compute coordinated samples $S^{(i)}$ for each $\mathrm{w}^{(\mathrm{i})}$

- multi-objective sample S is the union $\mathrm{S} = \cup_i S^{(i)}$
- Estimation: Inverse probability with sampling probabilities $p_x = \max_i p^{(i)}(x)$

Gains: storage (maximum overlap)

Higher accuracy that with using just the dedicated sample

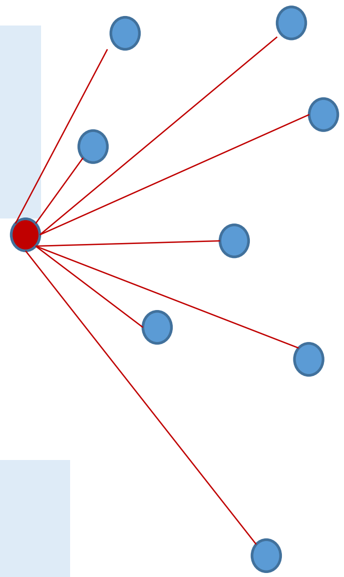
Multi-objective Sample

- Same keys can have different "weights:" IP flows have bytes, packets, count
- We want to answer queries with respect to all weights.
- **Naïve solution**: 3 disjoint samples
- **Smart solution**: A single multi-objective sample

“Classic” centrality, Coresets for Clustering

”Classic” centrality: Pointset X in a metric space

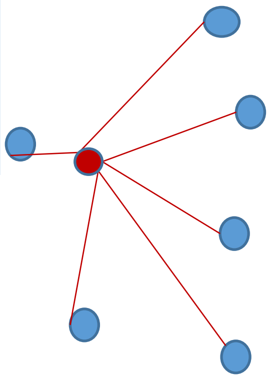
Query: point x estimate $\sum_{y \in X} d(x, y)$



- “Instance” for each point x with weights $d(x, y)$ for $y \in X$
- Soft PPS + multi-objective sample size $O(\epsilon^{-2})$

Clustering cost: Pointset X

Query: k-tuple $\boldsymbol{x} = (x_1, \dots, x_k)$ estimate $\sum_{y \in X} d(\boldsymbol{x}, y)$ where $d(\boldsymbol{x}, y) = \min_{i \in [k]} d(x_i, y)$



- “Instance” for each tuple $\boldsymbol{x}$ with weights $d(\boldsymbol{x}, y)$ for $y \in X$
- Soft PPS + multi-objective sample size $O(k\,\epsilon^{-2})$

Multi-objective sample of monotone weights

All weights that are ”ordered” the same way $w_1^{(i)} \geq \ w_1^{(i)} \geq w_2^{(i)} \geq w_3^{(i)} \geq$

Multi-objective sample has

- (expected) size: $O(k \ln n)$, where $n = $ #keys

Suffices to take union of coordinated samples of all **unweighted** prefixes of the order

Application: Data Streams time-decaying aggregations

monotone non-increasing $\alpha(x)$, and segment $H \subset V$

$$A_\alpha = \sum_{u \in H} \alpha\,(t_u)$$

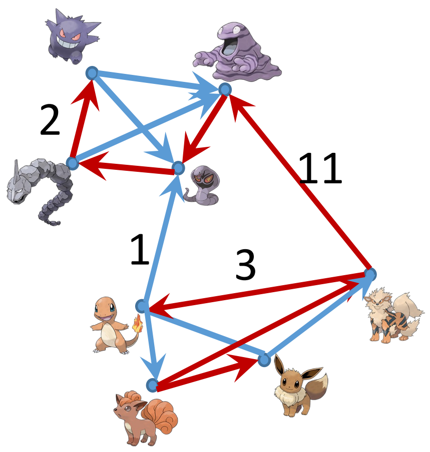
t_u: Elapsed time from start of stream to u

t_u: Elapsed time from u to current time

Graph Sketches: All-distance Sketches

Graph $G = (V, E)$ For each node v :
$\mathrm{ADS}(v)$: A union of coordinated samples of all its r-neighborhoods $N_r(v)$
$= \{u \mid d(v, u) \leq r\}$

Coordination across nodes and distances!



- **Small size**: $\mathrm{E}[\,|\mathrm{ADS}(v)|] = k \log(n)$
  - $\mathrm{ADS}(v)$ is multi-objective with monotone weights!

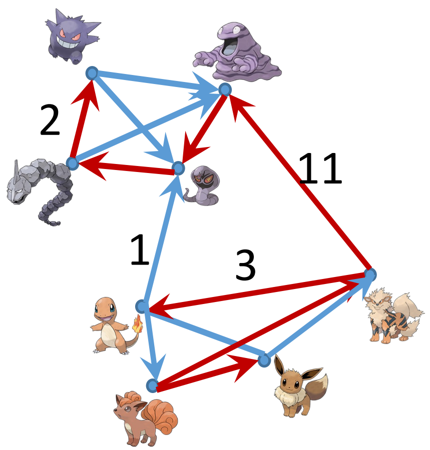
    When ordered by distance $(1, \ldots, 1, 0, \ldots, 0)$
- **Near-linear computation**/storage: $\tilde{O}(k|E|)$

Graph Sketches: All-distance Sketches…

Graph $G = (V, E)$ For each node v :
$\mathrm{ADS}(v)$: A union of coordinated samples of all its r-neighborhoods $N_r(v)$
$= \{u \mid d(v, u) \leq r\}$

Queries:

- **Node-centric queries**: centrality/kernel density,
- **Inter-node queries**: approximate distance oracles, similarity, influence (merged coverage of multiple nodes



Distance-decaying centrality query: (monotone weights!)

Query: Node v, monotone non-increasing $\alpha(x)$, selection predicate h

Estimate centrality (= kernel density) of v

$$C_\alpha(v, h) = \sum_u h(v) \cdot \alpha(d_{vu})$$

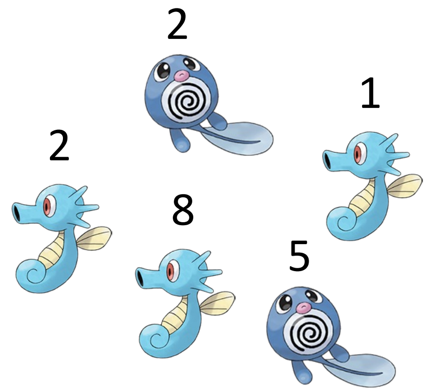
Big Ideas – Outline

- Composable summary structures
- PPS sampling for Linear Queries
- Sampling via Order Statistics
  - Distributed/streaming use
  - Graph sketches
  - Estimation of Linear Queries
  - Soft PPS
  - Sticky per-key randomness
    - Coordinated Samples
    - Multi-objective Samples

Sampling Unaggregated raw data

- Max aggregation
- Sum aggregation
- Functions of frequency
  - Transform to Max
  - Transform to Heavy Hitters

Unaggregated raw data



- Data element $e \in E$ has key and value *(e.key,e.value)*
- Multiple elements may share the same key

  - Max agg: $w_x = \max_{e|e.key=x} \mathrm{e.value}$

  - Sum agg: $w_x = \sum_{e|e.key=x} \mathrm{e.value}$

- Naïve: Aggregate pairs (x, w_x), then sample – requires structure size O(#distinct keys)
- Goal: Work over raw data via composable structures of size $O(k)$

Unaggregated data: Max-Distinct Sampling

- Max agg: $w_x = \max_{e|e.key=x} \mathrm{e.value}$

Use hash-based sticky per-key randomness $\mathrm{seed}(x) \sim \mathrm{Exp}[1]$

Locally map each element

$$e = (e.\mathrm{key}, e.\mathrm{val}) \mapsto e^* = (e.\mathrm{key}, \frac{seed(e.\mathrm{key})}{e.\mathrm{val}})$$

Aggregate mapped elements E^* to find k unique keys with lowest scores (via a composable bottom-k structure)

Correctness: the minimum value of a key in E^* corresponds to score of largest $e.\mathrm{val}$

Unaggregated data: Sum aggregation

- Sum agg: $w_x = \sum_{e|e.key=x} \text{e.value}$

Locally map each element

$$e = (e.\text{key}, e.\text{val}) \mapsto e^* = (e.\text{key}, v \sim \text{Exp}[e.val])$$

Aggregate mapped elements E^* via composable bottom-k structures
(keep lowest score for each key)

Correctness: the minimum value of a key x in E^* has distribution

$$\min_{\text{e}\in E \mid \text{e.key=x}} \text{Exp}[e.val] \equiv \text{Exp}\left[\sum_{e\in E \mid \text{e.key=x}} e.val\right]$$ (property of exp distribution)

Note: No sticky per-key randomness

Caveat: We have the right sample but we don't have weights w_x for sampled keys!!

Unaggregated data: Sum aggregation estimation via inversion

We have a PPSWOR sample with no weights w_x for sampled keys!!

Easy solution: Perform a second aggregation pass to sum weights of the sampled keys (simple composable structure of size k)

Solution for streaming: Can't collect weights for sampled keys. But can collect weights so that we have a handle on the distribution of the part we "missed". "Invert" that distribution to obtain estimates.

Surprise: We get the ~optimal error bounds on estimating linear statistics

Unaggregated data: Functions of frequency

Sum of weights ("frequency"): $\sum_{e|e.key=x} \mathrm{e.value}$

Monotone non-decreasing f

Our weights are function of frequency $w_x = f(\sum_{e|e.key=x} \mathrm{e.value})$

!! Some functions are hard: can't do super-quadratic growth

Two "transform" based ideas

- Via Max-Distinct sampling applicable to concave sublinear functions
- Via heavy hitters for moment functions $f(\mathrm{w}) = \mathrm{w}^{\mathrm{p}}$ $\mathrm{p} \in [0,2]$

Unaggregated data: Functions of frequency Transform to (Max)-Distinct

Concave sublinear functions are sub-linear with non-increasing growth. Examples:

$f(w) = w^p \ , p \in [0,1]$ (low frequency moments)

$f(w) = \min\{T, w\}$ (capping function)

$f(w) = \log(1 + w)$

High level idea: These functions are ``between" max and sum aggregations. We "mix" the element maps.

Properties:

- No sticky randomness
- Can get a multi-objective sample of all concave-sublinear functions (logarithmic factor increase in sample size)
- When only computing the statistics $\sum_x f(\sum_{e|e.key=x} \text{e. value})$ -- can "strip" the sample and get hyperloglog-like composable structures.

Unaggregated data: Functions of frequency Transform to heavy hitters

- $w_x = \left(\sum_{e|e.key=x} e.val\right)^{\mathrm{p}}$

Use hash-based sticky per-key randomness $\mathrm{seed}(x) \sim \mathrm{Exp}[1]$

Locally map each element

$$e = (e.\mathrm{key}, e.\mathrm{val}) \mapsto e^* = \left(e.\mathrm{key}, \frac{e.val}{\left(seed(e.key)\right)^{\frac{1}{p}}}\right)$$

Sum aggregate of E^* has weights $\frac{\sum_{e\,|\,e.key=x} e.val}{\left(seed(e.key)\right)^{\frac{1}{p}}}$. We want the top-k

top-k **of sum-aggs** generally needs structure size O(#distinct) !!
Fortunately, top-k are ℓ_p heavy-hitters. When $p \leq 2$, we apply a HH sketch (e.g. Count Sketch) to E^*

Summary: Big Ideas

- Many of the ideas originated in the statistics literature for very different and typically much smaller scale applications (survey sampling)
- These ideas, and their extensions, found and continue to find new applications
- See writeup for references and pointers

Thank you!