Tight Bounds for Single-Pass Streaming Complexity of the Set Cover Problem

Sepehr Assadi

University of Pennsylvania

Joint work with Sanjeev Khanna (Penn) and Yang Li (Penn)

The Set Cover Problem

- Input: A collection of m sets $S_1, \ldots, S_m$ from a universe $[n]$.
- Goal: Choose a smallest subset C of the sets from $S_1, \ldots, S_m$ such that C covers $[n]$, i.e., $\bigcup_{i \in C} S_i = [n]$.

The Set Cover Problem

- Input: A collection of m sets $S_1, \ldots, S_m$ from a universe $[n]$.
- Goal: Choose a smallest subset C of the sets from $S_1, \ldots, S_m$ such that C covers $[n]$, i.e., $\bigcup_{i \in C} S_i = [n]$.

The sets maybe weighted in general.

We use OPT to denote the optimal solution size/weight.

The Set Cover Problem

- Input: A collection of m sets $S_1, \ldots, S_m$ from a universe $[n]$.
- Goal: Choose a smallest subset C of the sets from $S_1, \ldots, S_m$ such that C covers $[n]$, i.e., $\bigcup_{i \in C} S_i = [n]$.

The sets maybe weighted in general.

We use OPT to denote the optimal solution size/weight.

Approximation vs Estimation:

- α-approximation: output a set cover of size at most $\alpha \cdot \mathsf{OPT}$ plus a certificate of coverage for each element $e \in [n]$.
- α-estimation: output an estimate for the size of minimum set cover in range $[\mathsf{OPT}, \alpha \cdot \mathsf{OPT}]$.

The Set Cover Problem

- A classic optimization problem with many applications.

The Set Cover Problem

- A classic optimization problem with many applications.

- A well-understood problem in the classical setting:
  - Admits a poly-time greedy $\ln n$-approximation algorithm.
  - No poly-time $(1-\epsilon) \cdot \ln n$-estimation algorithm unless $\mathsf{P} = \mathsf{NP}$.

The Set Cover Problem

- A classic optimization problem with many applications.

- A well-understood problem in the classical setting:
  - Admits a poly-time greedy $\ln n$-approximation algorithm.
  - No poly-time $(1-\epsilon) \cdot \ln n$-estimation algorithm unless P = NP.

- This talk: space complexity of approximating the set cover problem in the streaming model.

The Streaming Set Cover Problem

Model:

- The input sets $S_1, \ldots, S_m$ are presented one by one in a stream.
- The streaming algorithm has a small space to maintain a summary of the input sets.
- At the end, the algorithm outputs an exact/approximate set cover using this summary.

The Streaming Set Cover Problem

Model:

- The input sets $S_1, \ldots, S_m$ are presented one by one in a stream.
- The streaming algorithm has a small space to maintain a summary of the input sets.
- At the end, the algorithm outputs an exact/approximate set cover using this summary.

Introduced originally by [SG09] and further studied in several recent works [ER14, DIMV14, IMV15, CW16, HPIMV16].

The Streaming Set Cover Problem

Model:

- The input sets $S_1, \ldots, S_m$ are presented one by one in a stream.
- The streaming algorithm has a small space to maintain a summary of the input sets.
- At the end, the algorithm outputs an exact/approximate set cover using this summary.

Introduced originally by [SG09] and further studied in several recent works [ER14, DIMV14, IMV15, CW16, HPIMV16].

Remark. We are not concerned with poly-time computability in this model.

State of the Art for Single-Pass Algorithms

Result	Space	Performance Ratio
Exact	$O(mn)$	1
[IMV15]	$\Omega(mn)$	$3/2 - \epsilon$
[ER14]	$O(n)$	$O(\sqrt{n})$
[ER14]	$\Omega(m)$	$o(\sqrt{n})$

State of the Art for Single-Pass Algorithms

Result	Space	Performance Ratio
Exact	$O(mn)$	1
[IMV15]	$\Omega(mn)$	$3/2-\epsilon$
[ER14]	$O(n)$	$O(\sqrt{n})$
[ER14]	$\Omega(m)$	$o(\sqrt{n})$

Many known results for multi-pass algorithms as well: [SG09, IMV15, CW16] ...

State of the Art for Single-Pass Algorithms

Result	Space	Performance Ratio
Exact	$O(mn)$	1
[IMV15]	$\Omega(mn)$	$3/2 - \epsilon$
[ER14]	$O(n)$	$O(\sqrt{n})$
[ER14]	$\Omega(m)$	$o(\sqrt{n})$

Single-pass Algorithms:

- $o(m)$ space regime is settled by the results of [ER14].
- However, sublinear space regime, that is, what can be done in $o(mn)$ space is wide open.

State of the Art for Single-Pass Algorithms

Result	Space	Performance Ratio
Exact	$O(mn)$	1
[IMV15]	$\Omega(mn)$	$3/2 - \epsilon$
[ER14]	$O(n)$	$O(\sqrt{n})$
[ER14]	$\Omega(m)$	$o(\sqrt{n})$

Single-pass Algorithms:

- $o(m)$ space regime is settled by the results of [ER14].
- However, sublinear space regime, that is, what can be done in $o(mn)$ space is wide open.
  - For example, is $O(1)$ approximation possible in $o(mn)$ space?
  - In general, what is the space-approximation tradeoff in this regime?

Our First Result

A tight space-approximation tradeoff for single-pass streaming algorithms:

Theorem

For any $\alpha = o(\sqrt{n})$, $\widetilde{\Theta}(mn/\alpha)$ *space is both sufficient and necessary for* α*-approximating the set cover problem.*

α-Approximation in $\widetilde{O}(mn/\alpha)$ space

A simple algorithm for (weighted) set cover:

1. Guess OPT and ignore sets with weight $>$ OPT.

α-Approximation in $\widetilde{O}(mn/\alpha)$ space

A simple algorithm for (weighted) set cover:

1. Guess OPT and ignore sets with weight $>$ OPT.
2. Prune: Include a set if it covers more than n/α new elements and remove these elements from the universe.

   (at most α sets would be included with total weight $\leq \alpha \cdot$ OPT)

α-Approximation in $\widetilde{O}(mn/\alpha)$ space

A simple algorithm for (weighted) set cover:

1. Guess OPT and ignore sets with weight $>$ OPT.
2. Prune: Include a set if it covers more than n/α new elements and remove these elements from the universe.

   (at most α sets would be included with total weight $\leq \alpha \cdot$ OPT)
3. Store all remaining sets over the new universe.

   (each remaining set contains $< n/\alpha$ elements and hence they can all be stored in $O(mn/\alpha)$ space)

α-Approximation in $\widetilde{O}(mn/\alpha)$ space

A simple algorithm for (weighted) set cover:

1. Guess OPT and ignore sets with weight $>$ OPT.
2. Prune: Include a set if it covers more than n/α new elements and remove these elements from the universe.

   (at most α sets would be included with total weight $\leq \alpha \cdot \text{OPT}$)
3. Store all remaining sets over the new universe.

   (each remaining set contains $< n/\alpha$ elements and hence they can all be stored in $O(mn/\alpha)$ space)
4. Solve the store set cover instance optimally to cover the elements remained uncovered by the prune step.

α-Approximation in $\widetilde{O}(mn/\alpha)$ space

A simple algorithm for (weighted) set cover:

1. Guess OPT and ignore sets with weight $>$ OPT.
2. Prune: Include a set if it covers more than n/α new elements and remove these elements from the universe.

   (at most α sets would be included with total weight $\leq \alpha \cdot$ OPT)
3. Store all remaining sets over the new universe.

   (each remaining set contains $< n/\alpha$ elements and hence they can all be stored in $O(mn/\alpha)$ space)
4. Solve the store set cover instance optimally to cover the elements remained uncovered by the prune step.

Our lower bound shows that this simple algorithm is essentially the best possible in terms of space requirement!

Approximation vs Estimation

Previous upper bounds are for the approximation problem, while lower bounds are for estimation.

Approximation vs Estimation

Previous upper bounds are for the approximation problem, while lower bounds are for estimation.

However, our $\Omega(mn/\alpha)$ lower bound strongly relies on the fact that we are solving the approximation problem and not simply estimating the value of the optimal set cover.

Approximation vs Estimation

Previous upper bounds are for the approximation problem, while lower bounds are for estimation.

However, our $\Omega(mn/\alpha)$ lower bound strongly relies on the fact that we are solving the approximation problem and not simply estimating the value of the optimal set cover.

Question: Can it be that estimation is strictly easier than approximation?

Our Second Result

Estimation is indeed distinctly easier!

Theorem

For any $\alpha = o(\sqrt{n})$, there exists a randomized α-estimation $\widetilde{O}(mn/\alpha^2)$ space algorithm for the streaming set cover problem.

Works in general for any covering integer program, and in particular for weighted set-cover or set multi-cover problem.

Our Third Result

The factor α gap between space requirements of approximation versus estimation algorithms for streaming set cover is tight.

Theorem

For any $\alpha = o(\sqrt{n})$, any randomized algorithm that α-estimates the set cover problem requires $\widetilde{\Omega}(mn/\alpha^2)$ space.

This lower bound holds even for random arrival streams.

$\Omega(mn/\alpha)$ Space is Necessary to Compute an α-Approximate Set Cover

Communication Complexity

We use communication complexity paradigm to prove our lower bound.

Communication Complexity

We use communication complexity paradigm to prove our lower bound.

One-way Two-player Communication Model:

- Alice gets a private input X and Bob gets a private input Y.
- Their goal is to compute a function $P(X,Y)$.
- Alice is allowed to send a single message M to Bob.
- Bob uses the message M plus his input to compute $f(M,Y) \approx P(X,Y)$.

Communication Complexity

We use communication complexity paradigm to prove our lower bound.

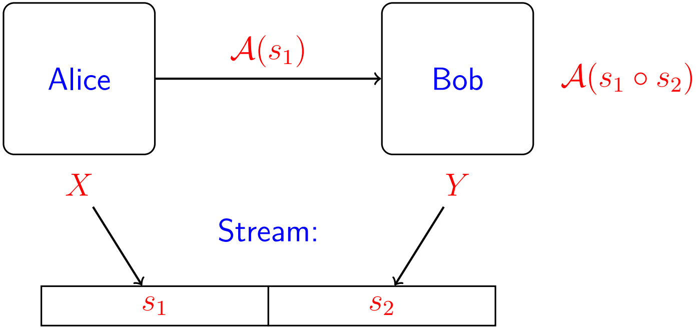
One-way Two-player Communication Model:

- Alice gets a private input X and Bob gets a private input Y.
- Their goal is to compute a function $P(X, Y)$.
- Alice is allowed to send a single message M to Bob.
- Bob uses the message M plus his input to compute $f(M, Y) \approx P(X, Y)$.

Communication Complexity $\mathsf{CC}(P)$: the minimum length of a message for any protocol that solves P with probability at least $2/3$.

Connection to Streaming Complexity

Space needed by any streaming algorithm for a problem P is at least the communication complexity of P.



A Hard Input Distribution for Set Cover

Theorem

$$\textit{CC}(\alpha\textit{-Approximate Set Cover}) = \Omega(mn/\alpha)$$

A Hard Input Distribution for Set Cover

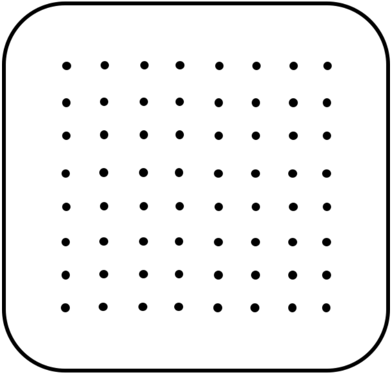
Theorem

$$\textit{CC}(\alpha\textit{-Approximate Set Cover}) = \Omega(mn/\alpha)$$

- Alice and Bob each gets a collection of sets.
- Alice sends a single message to Bob and Bob outputs an α-approximate set cover.

A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

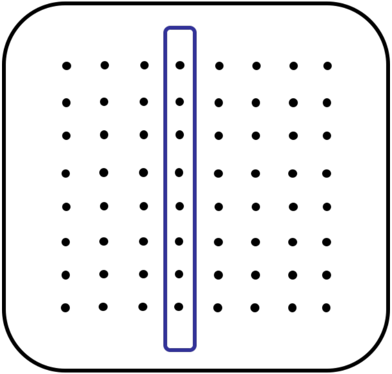


$[n]$

A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

- Alice: near orthogonal sets of size n/α.

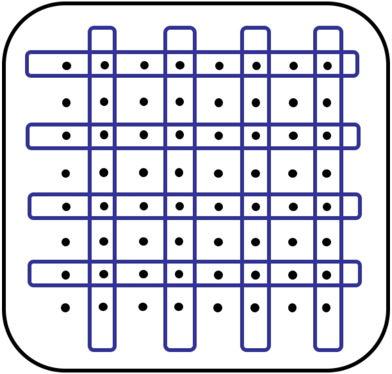


$[n]$

A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

- Alice: near orthogonal sets of size n/α.

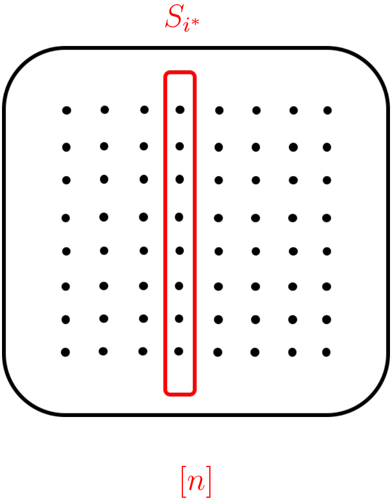


$[n]$

A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

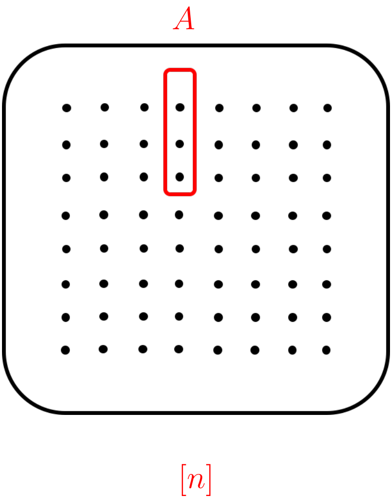
- Alice: near orthogonal sets of size n/α.
- Bob: a single set T of size $n - 6\alpha$:



A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

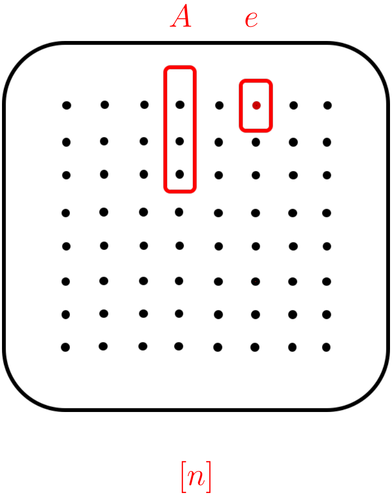
- Alice: near orthogonal sets of size n/α.
- Bob: a single set T of size $n - 6\alpha$.



A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

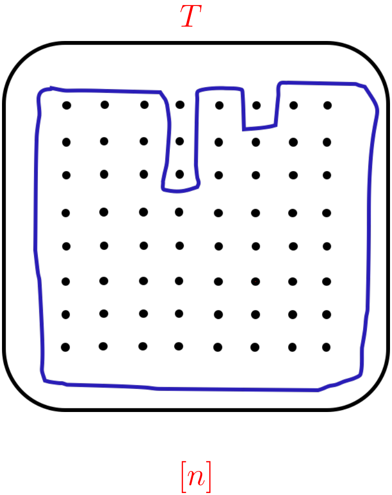
- Alice: near orthogonal sets of size n/α.
- Bob: a single set T of size $n - 6\alpha$.



A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

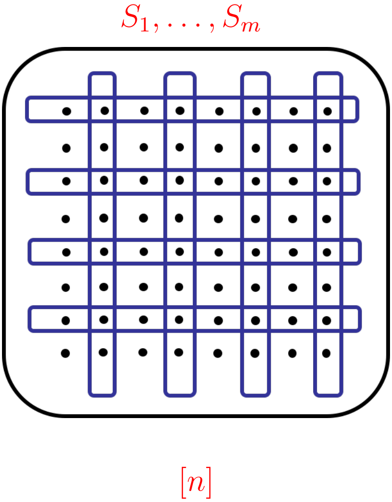
- Alice: near orthogonal sets of size n/α.
- Bob: a single set T of size $n - 6\alpha$.



A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

- Alice: a collection of m sets $S_1, \ldots, S_m$.

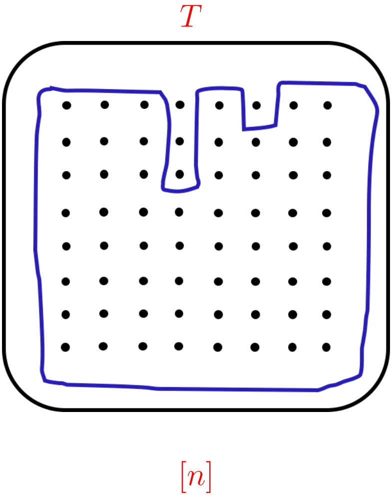


A Hard Input Distribution for Set Cover

Input Distribution $\mathcal{D}$:

- Alice: a collection of m sets $S_1, \ldots, S_m$.

- Bob: a single set T.



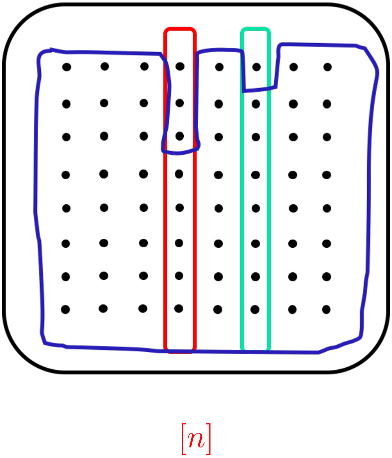
A Hard Input Distribution for Set Cover

The optimal set cover size is at most 3:

A Hard Input Distribution for Set Cover

The optimal set cover size is at most 3:

Use T,S_{i^*}, and one more set for covering the special element.



Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

Solving set cover on $\mathcal{D}$ is equivalent to identifying the special element.

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

Solving set cover on $\mathcal{D}$ is equivalent to identifying the special element.

1. Bob can identify the set S_{i^*} with small communication.

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

Solving set cover on $\mathcal{D}$ is equivalent to identifying the special element.

1. Bob can identify the set S_{i^*} with small communication.
2. Bob knows using T and S_{i^*} he can cover all but a single element, i.e., the special element e.

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

Solving set cover on $\mathcal{D}$ is equivalent to identifying the special element.

1. Bob can identify the set S_{i^*} with small communication.
2. Bob knows using T and S_{i^*} he can cover all but a single element, i.e., the special element e.
3. Bob's task is then to identify the special element in $\overline{T}$.

   Identify $=$ find a small enough subset of $\overline{T}$ that contains e.

   In other words, trap the special element e.

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

Solving set cover on $\mathcal{D}$ is equivalent to identifying the special element.

1. Bob can identify the set S_{i^*} with small communication.
2. Bob knows using T and S_{i^*} he can cover all but a single element, i.e., the special element e.
3. Bob's task is then to identify the special element in $\overline{T}$.

   Identify = find a small enough subset of $\overline{T}$ that contains e.
   In other words, trap the special element e.
4. Bob can then cover the trap-set using sets other than S_{i^*}.

Proof Sketch

How small is small enough for the trap-set size?

Proof Sketch

How small is small enough for the trap-set size?

1. Optimal set cover size is at most 3, hence Bob is allowed to use up to 3α sets in the set cover.

Proof Sketch

How small is small enough for the trap-set size?

1. Optimal set cover size is at most 3, hence Bob is allowed to use up to 3α sets in the set cover.
2. The trap-set needs to be coverable by $< 3\alpha$ sets other than S_{i^*}.

Proof Sketch

How small is small enough for the trap-set size?

1. Optimal set cover size is at most 3, hence Bob is allowed to use up to 3α sets in the set cover.
2. The trap-set needs to be coverable by $< 3\alpha$ sets other than S_{i^*}.
3. The near orthogonality of the sets implies that the trap-set has to be of size $< 3\alpha$.

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

Suppose Alice only has a single set, i.e., only S_{i^*}; then, trapping the special element requires full knowledge of Alice's set.

Trap problem: the communication problem of trapping the special element, when Alice has a single set S and Bob has a single set $A \cup \{e\}$.

Proof Sketch

Lemma

$$CC(\text{Trap}) = \Omega(n/\alpha)$$

Proof Sketch

Lemma

$$\mathsf{CC}(\mathsf{Trap}) = \Omega(n/\alpha)$$

Intuitively,

1. If Alice sends $o(n/\alpha)$ bits, only $o(1)$ fraction of the set S is revealed to Bob.

Proof Sketch

Lemma

$$CC(\text{Trap}) = \Omega(n/\alpha)$$

Intuitively,

1. If Alice sends $o(n/\alpha)$ bits, only $o(1)$ fraction of the set S is revealed to Bob.
2. Since A is chosen uniformly at random from S, Bob can only determine $o(1)$ fraction of A that belongs to S.

Proof Sketch

Lemma

$$\mathit{CC}(\mathsf{Trap}) = \Omega(n/\alpha)$$

Intuitively,

1. If Alice sends $o(n/\alpha)$ bits, only $o(1)$ fraction of the set S is revealed to Bob.
2. Since A is chosen uniformly at random from S, Bob can only determine $o(1)$ fraction of A that belongs to S.
3. Consequently, Bob can only trap the special element by a set of size $(1 - o(1))\,|A| > 3\alpha$.

Proof Sketch

Lemma

$$CC(\text{Trap}) = \Omega(n/\alpha)$$

Intuitively,

1. If Alice sends $o(n/\alpha)$ bits, only $o(1)$ fraction of the set S is revealed to Bob.
2. Since A is chosen uniformly at random from S, Bob can only determine $o(1)$ fraction of A that belongs to S.
3. Consequently, Bob can only trap the special element by a set of size $(1 - o(1)) |A| > 3\alpha$.

We formalize this using an information-theoretic argument and a novel reduction from the Index problem.

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

When i^* is not known to Alice, trapping the special element requires m times more communication:

$$CC(\alpha\text{-Approximate Set Cover}) \approx m \cdot CC(\text{Trap})$$

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

When i^* is not known to Alice, trapping the special element requires m times more communication:

$$CC(\alpha\text{-Approximate Set Cover}) \approx m \cdot CC(\text{Trap})$$

Intuitively,

1. The index i^* is unknown to Alice, hence Alice's message essentially needs to solve Trap for most indices $i \in [m]$.

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

When i^* is not known to Alice, trapping the special element requires m times more communication:

$$CC(\alpha\text{-Approximate Set Cover}) \approx m \cdot CC(\text{Trap})$$

Intuitively,

1. The index i^* is unknown to Alice, hence Alice's message essentially needs to solve Trap for most indices $i \in [m]$.
2. The sets are chosen independently, hence information sent for one set cannot be used for solving Trap on another set.

Proof Sketch

Why $\mathcal{D}$ is a hard distribution?

Claim

When i^* is not known to Alice, trapping the special element requires m times more communication:

$$\mathit{CC}(\alpha\text{-}\mathit{Approximate\ Set\ Cover}) \approx m \cdot \mathit{CC}(\mathrm{Trap})$$

Intuitively,

1. The index i^* is unknown to Alice, hence Alice's message essentially needs to solve Trap for most indices $i \in [m]$.
2. The sets are chosen independently, hence information sent for one set cannot be used for solving Trap on another set.

We formalize this using information complexity and a direct-sum style argument.

Summary

Hence,

$$\text{CC}(\alpha\text{-Approximate Set Cover}) \approx \Omega(mn/\alpha)$$

Communication complexity is also a lower bound on the space complexity of the streaming algorithms:

Theorem

For any $\alpha = o(\sqrt{n})$, $\Omega(mn/\alpha)$ space is necessary for α-approximating the set cover problem.

Moreover, this space-approximation tradeoff is tight.

$\widetilde{O}(mn/\alpha^2)$ Space is Sufficient for α-Estimating Set Cover

An α-Estimation Algorithm in $\widetilde{O}(mn/\alpha^2)$ Space

We show that,

Theorem

There exists a single-pass streaming that α-estimates the weighted set cover problem in $\widetilde{O}(mn/\alpha^2)$ space.

These ideas can be further generalized to estimate optimal solution value of any covering integer program.

α-Approximation in $\widetilde{O}(mn/\alpha)$ space

A simple algorithm for (weighted) set cover:

1. Guess OPT and ignore sets with weight $>$ OPT.
2. Prune: Include a set if it covers more than n/α new elements and remove these elements from the universe.

   (at most α sets would be included with total weight $\leq \alpha \cdot$ OPT)
3. Store all remaining sets over the new universe.

   (each remaining set contains $< n/\alpha$ elements and hence they can all be stored in $O(mn/\alpha)$ space)
4. Solve the store set cover instance optimally to cover the elements remained uncovered by the prune step.

Element Sampling

How to save another factor α to achieve $O(mn/\alpha^2)$ when the goal is only estimating?

Element Sampling

How to save another factor α to achieve $O(mn/\alpha^2)$ when the goal is only estimating?

Element Sampling:

- Sample each element with probability $1/\alpha$ and work with the sampled universe in the second phase of the algorithm.
- Store the sampled instance completely (after pruning).

  (each set has $\leq n/\alpha^2$ elements in the sampled universe and hence total space requirement is $O(mn/\alpha^2)$)

The hope is that the sampling procedure reduces the weight of the optimal set cover by a factor of at most α.

Element Sampling

Let

- $\mathcal{I}$ be an instance of the weighted set cover problem.
- $\mathcal{I}_\alpha$ be an instance obtained from $\mathcal{I}$ by sampling each element of the universe $[n]$ with probability $1/\alpha$.

Element Sampling

Let

- $\mathcal{I}$ be an instance of the weighted set cover problem.
- $\mathcal{I}_\alpha$ be an instance obtained from $\mathcal{I}$ by sampling each element of the universe $[n]$ with probability $1/\alpha$.

Clearly, $\mathsf{OPT}(\mathcal{I}_\alpha) \leq \mathsf{OPT}(\mathcal{I})$.

Element Sampling

Let

- $\mathcal{I}$ be an instance of the weighted set cover problem.
- $\mathcal{I}_\alpha$ be an instance obtained from $\mathcal{I}$ by sampling each element of the universe $[n]$ with probability $1/\alpha$.

Clearly, $\mathsf{OPT}(\mathcal{I}_\alpha) \leq \mathsf{OPT}(\mathcal{I})$.

Ideally, we also want $\mathsf{OPT}(\mathcal{I}_\alpha) \geq \mathsf{OPT}(\mathcal{I})/\alpha$ with probability $\Omega(1)$.

This way, we can use $\mathsf{OPT}(\mathcal{I}_\alpha)$ as a proxy for $\mathsf{OPT}(\mathcal{I})$.

Element Sampling

Let

- $\mathcal{I}$ be an instance of the weighted set cover problem.
- $\mathcal{I}_\alpha$ be an instance obtained from $\mathcal{I}$ by sampling each element of the universe $[n]$ with probability $1/\alpha$.

Clearly, $\mathsf{OPT}(\mathcal{I}_\alpha) \leq \mathsf{OPT}(\mathcal{I})$.

Ideally, we also want $\mathsf{OPT}(\mathcal{I}_\alpha) \geq \mathsf{OPT}(\mathcal{I})/\alpha$ with probability $\Omega(1)$.

This way, we can use $\mathsf{OPT}(\mathcal{I}_\alpha)$ as a proxy for $\mathsf{OPT}(\mathcal{I})$.

But is this true?

Element Sampling

This is not true in general.

Consider the following instance $\mathcal{I}$ with n sets:

- $S_1 = \{1\}$ with weight $W \gg n$.
- $S_i = \{i\}$ for $i > 1$ with weight 1.

Clearly,

- $\mathsf{OPT}(\mathcal{I}) = (n-1) + W$
- $\Pr\left[\mathsf{OPT}(\mathcal{I}_\alpha) \geq \mathsf{OPT}(\mathcal{I})/\alpha\right] = o(1)$

Element Sampling

This is not true in general.

Consider the following instance $\mathcal{I}$ with n sets:

- $S_1 = \{1\}$ with weight $W \gg n$.
- $S_i = \{i\}$ for $i > 1$ with weight 1.

Clearly,

- $\mathsf{OPT}(\mathcal{I}) = (n-1) + W$
- $\mathsf{Pr}\left[\mathsf{OPT}(\mathcal{I}_\alpha) \geq \mathsf{OPT}(\mathcal{I})/\alpha\right] = o(1)$

The problem is existence of elements that are too expensive to cover.

Element Sampling Lemma

- For each element $e \in [n]$, define $\mathit{Cost}(e)$ to be the minimum weight of any set that covers e.
- Define $\mathit{Cost}(\mathcal{I}) := \max_{e \in [n]} \mathit{Cost}(e)$.

Element Sampling Lemma

- For each element $e \in [n]$, define $\mathit{Cost}(e)$ to be the minimum weight of any set that covers e.
- Define $\mathit{Cost}(\mathcal{I}) := \max_{e \in [n]} \mathit{Cost}(e)$.

$\mathit{Cost}(\mathcal{I})$ is clearly a lower bound on $\mathsf{OPT}(\mathcal{I})$.

Element Sampling Lemma

- For each element $e \in [n]$, define $\mathit{Cost}(e)$ to be the minimum weight of any set that covers e.
- Define $\mathit{Cost}(\mathcal{I}) := \max_{e \in [n]} \mathit{Cost}(e)$.

$\mathit{Cost}(\mathcal{I})$ is clearly a lower bound on $\mathrm{OPT}(\mathcal{I})$.

Lemma (Element Sampling Lemma)

For any instance $\mathcal{I}$, let $\mathcal{I}_\alpha$ be an instance obtained by sampling each element independently with probability $\frac{\ln(n)}{\alpha}$, then,

$$\Pr\left[\mathit{OPT}(\mathcal{I}_\alpha) + \mathit{Cost}(\mathcal{I}) \geq \frac{\mathit{OPT}(\mathcal{I})}{\alpha}\right] \geq \frac{1}{2}$$

Upper Bound Statement

Theorem

For any $\alpha = o(\sqrt{n})$, $\widetilde{\Theta}(mn/\alpha^2)$ space is sufficient for α-estimating the weighted set cover problem.

Moreover, this space-estimation tradeoff is tight.

Summary of Our Results

For the set cover problem in single-pass streams,

Summary of Our Results

For the set cover problem in single-pass streams,

α-approximation:

$\widetilde{\Theta}(mn/\alpha)$ space is necessary and sufficient.

Summary of Our Results

For the set cover problem in single-pass streams,

α-approximation:

$\widetilde{\Theta}(mn/\alpha)$ space is necessary and sufficient.

α-estimation:

$\widetilde{\Theta}(mn/\alpha^2)$ space is necessary and sufficient.

Summary of Our Results

For the set cover problem in single-pass streams,

α-approximation:

$\widetilde{\Theta}(mn/\alpha)$ space is necessary and sufficient.

α-estimation:

$\widetilde{\Theta}(mn/\alpha^2)$ space is necessary and sufficient.

Our results resolve the space-complexity of set cover in single-pass streams.

Questions?

Amit Chakrabarti and Anthony Wirth.
Incidence geometries and the pass complexity of semi-streaming set cover.
In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 1365–1373, 2016.

Erik D. Demaine, Piotr Indyk, Sepideh Mahabadi, and Ali Vakilian.
On streaming and communication complexity of the set cover problem.
In *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, pages 484–498, 2014.

Yuval Emek and Adi Rosén.
Semi-streaming set cover - (extended abstract).

In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 453–464, 2014.

Sariel Har-Peled, Piotr Indyk, Sepideh Mahabadi, and Ali Vakilian.
Towards tight bounds for the streaming set cover problem.
To appear in PODS, 2016.

Piotr Indyk, Sepideh Mahabadi, and Ali Vakilian.
Towards tight bounds for the streaming set cover problem.
CoRR, abs/1509.00118, 2015.

Noam Nisan.
The communication complexity of approximate set packing and covering.
In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, pages 868–875, 2002.

Barna Saha and Lise Getoor.
On maximum coverage in the streaming model & application to multi-topic blog-watch.
In *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA*, pages 697–708, 2009.