Fractional Set Cover in the Streaming Model

Piotr Indyk
MIT

Sepideh Mahabadi
Columbia Univ

Ronitt Rubinfeld
MIT/TAU

Jonathan Ullman
Northeastern Univ

Ali Vakilian
MIT

Anak Yodpinyanee
MIT
Set Cover Problem

Input: a collection $\mathcal{F}$ of sets $S_1, \ldots, S_m$ subset of $\mathcal{U} = \{1, \ldots, n\}$
  • i.e., $S_1 \cup S_2 \cup \cdots \cup S_m = \mathcal{U}$

Output: a subset $\mathcal{C}$ of $\mathcal{F}$ such that:
  • $\mathcal{C}$ covers $\mathcal{U}$
  • $|\mathcal{C}|$ is minimized

Complexity:
  • NP-hard
  • Greedy $(\ln n)$-approximation algorithm
  • Can’t do better unless P=NP [LY91][RS97][Fei98][AMS06][DS14]
Streaming Set Cover

• Model [SG09]
  • Sequential access to $S_1, S_2, ..., S_m$
  • One (or few) passes, sublinear (i.e., $o(mn)$) storage
  • (Hopefully) decent approximation factor

• Why?
  • A classic optimization problem
  • Application in “Big Data”: Clustering, Topic Coverage
  • One of few NP-hard problems studied in streaming
    • Other examples: Clustering, Max-Cut, Sub-Modular Optimization
Fractional Set Cover

- Each set can be picked fractionally (Assigning value $x_i$ to each set $S_i$)

  Fractional Solution of Set Cover $(x_1, x_2, \ldots, x_m)$

  Randomized Rounding

  $O(\log n)$-approximate Integral Solution

  Pick $S_i$ w.p. $\propto x_i \log n$

- The first step in solving covering LPs in stream
  - Packing LP (Fractional Maximum Matching) [AG11]
## Previous and Our Results

<table>
<thead>
<tr>
<th>INTEGRAL SET COVER</th>
<th>Approximation</th>
<th>Passes</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy Algorithm</td>
<td>$O(\log n)$</td>
<td>1</td>
<td>$O(mn)$</td>
</tr>
<tr>
<td></td>
<td>$O(\log n)$</td>
<td>$n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>[SG09]</td>
<td>$O(\log n)$</td>
<td>$O(\log n)$</td>
<td>$\tilde{O}(n)$</td>
</tr>
<tr>
<td>[ER14, CW16]</td>
<td>$O(n^\delta/\delta)$</td>
<td>$1/\delta - 1$</td>
<td>$\tilde{O}(n)$</td>
</tr>
<tr>
<td></td>
<td>$\Omega(n^\delta/\delta^2)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[DIMV14, HIMV16, BEM17]</td>
<td>$O(\rho/\delta)$</td>
<td>$O(1/\delta)$</td>
<td>$\tilde{O}(mn^\delta)$</td>
</tr>
<tr>
<td>[AKL16, A17]</td>
<td>$1/\delta$</td>
<td>polylog</td>
<td>$\tilde{\Omega}(mn^\delta)$</td>
</tr>
</tbody>
</table>

| FRACTIONAL SET COVER | $1 + \varepsilon$ | $O(1/\delta)$ | $\tilde{O}(mn^{0(\delta/\varepsilon)})$ |

**Assumption:** $m \gg n$

$\rho = \text{approximation factor for offline Set Cover}$

$\tilde{O}(f(m,n)) = O(f(m,n) \varepsilon^{-c} \log^c m \log^c n)$

$n = \text{number of elements}$

$m = \text{number of sets}$
Multiplicative Weight Update (Covering LP)

\[ w^1 \leftarrow (1, \cdots, 1) \quad \triangleright \text{uniform weights} \]

\textbf{For } t = 1, t \leq T \textbf{ do} \quad \triangleright \text{T iterations} 

\textbf{CoveringLP}(A_{n \times m}, c_m, b_n)

\text{Min} \quad c^T x \\
A x \geq b \\
x \geq 0
Multiplicative Weight Update (Covering LP)

\[ w^1 \leftarrow (1, \ldots, 1) \]

For \( t = 1, t \leq T \) do

\[ x^t \leftarrow \text{solution of Oracle} \]

\[ w^{t+1} \leftarrow \text{Update}(w^t, x^t) \]

\[ \bar{x} = \text{avg}(x_1, \ldots x_T) \]

Oracle \((A_{n \times m}, c, b, p^t)\)

\[
\begin{align*}
\text{Min} & \quad c^T x \\
(\mathbf{w}^t)^T A x & \geq (\mathbf{w}^t)^T b \\
x & \geq 0
\end{align*}
\]

MWU Update Rule:

\[
\begin{align*}
w_{e_i}^{t+1} & \equiv w_e^t \left( 1 - \varepsilon / \phi (A_e x^t - b_e) \right) \\
\forall i, t: & \quad -\phi \leq A_e x^t - b_e \leq \phi \\
\forall i: & \quad A_e \bar{x} \geq b_e - \varepsilon
\end{align*}
\]

How large should \( T \) be?

- Quality of oracle’s solution \( x^t \): (width of solution \( \phi \)):
  - or \( \max \) amount by which a constraint is over/under satisfied
- Quality of the desired solution (\( \varepsilon \)-feasible solution)

\[ T = O\left( \frac{\phi \log n}{\varepsilon^2} \right) \]
Multiplicative Weight Update (Set Cover)

**SET-COVER LP**(\(F, \mathcal{U}\)):

\[
\begin{align*}
\text{Min} & \quad \sum_{S \in F} x_S \\
\text{s.t.} & \quad \sum_{e \in S} x_S \geq 1 & \forall e \in \mathcal{U} \\
& \quad x_S \geq 0 & \forall S \in F
\end{align*}
\]
Multiplicative Weight Update (Set Cover)

**Feasibility-SET-COVER LP**\((\mathcal{F}, \mathcal{U}, k)\)

\[
\sum_{S \in \mathcal{F}} x_S \leq k
\]

\[\text{s.t. } \sum_{S: e \in S} x_S \geq 1 \quad \forall e \in \mathcal{U} \]
\[x_S \geq 0 \quad \forall S \in \mathcal{F}\]
Feasibility-SET-COVER LP($\mathcal{F}, \mathcal{U}, k$)

\[
\sum_{S \in \mathcal{F}} x_S \leq k
\]

\text{s.t.}
\[
\begin{align*}
\sum_{S : e \in S} x_S &\geq 1 & \forall e \in \mathcal{U} \\
x_S &\geq 0 & \forall S \in \mathcal{F}
\end{align*}
\]

Assign weight $w_e$ to each element $e$ (initially one)

Solve the \textit{weighted} average constraint \textit{approximately}!
Multiplicative Weight Update (Set Cover)

**Feasibility-SET-COVER LP** ($\mathcal{F}, \mathcal{U}, k$)

\[
\sum_{S \in \mathcal{F}} x_S \leq k
\]

\[
\sum_{e \in \mathcal{U}} w_e \sum_{S \in \mathcal{F}} x_S \geq \sum_{e \in \mathcal{U}} w_e \quad x_S \geq 0 \quad \forall S \in \mathcal{F}
\]

Assign weight $w_e$ to each element $e$ (initially one)  
Solve the *weighted* average constraint approximately!

Define

\[
\Sigma_{S \in \mathcal{F}} x_S w_s \geq \Sigma_{e \in \mathcal{U}} w_e \quad \Sigma_{S \in \mathcal{F}} x_S \geq \Sigma_{e \in \mathcal{U}} w_e \quad \Sigma_{S \in \mathcal{F}} x_S \leq k
\]

By *normalizing* weight vector $w$ (prob. vector $p$):

\[
\Sigma_{S \in \mathcal{F}} x_S p_S \geq 1
\]
Multiplicative Weight Update (Set Cover)

\[
\text{Oracle}(\mathcal{F}, \mathcal{U}, k, p)
\]

\[
\sum_{S \in \mathcal{F}} x_S \leq k
\]

\[
\sum_{S \in \mathcal{F}} x_S p_S \geq 1
\]

\[
x_S \geq 0 \quad \forall S \in \mathcal{F}
\]

Assign weight \(w_e\) to each element \(e\) (initially one)

Solve the \textit{weighted} average constraint \textit{approximately}!
Multiplicative Weight Update (Set Cover)

**Oracle**($\mathcal{F}, \mathcal{U}, k, p$)

\[
\sum_{S \in \mathcal{F}} x_S \leq k
\]

\[
\sum_{S \in \mathcal{F}} x_S p_S \geq 1 - \varepsilon
\]

\[
x_S \geq 0 \quad \forall S \in \mathcal{F}
\]

\[
-\phi \leq \sum_{S, e \in S} x_S - 1 \leq \phi \quad \forall e \in \mathcal{U}
\]

**T times**

Assign weight $w_e$ to each element $e$ (initially one)

Solve the *weighted* average constraint approx. w.r.t $p^t \propto w^t$: $x^t$

Update the prob vector

\[
p_{e, t+1} = p_{e, t}(1 - O(\varepsilon) \times (\sum x_{S, t}^t - 1))
\]

**MWU Theorem.** After $T = O(\frac{\phi \log n}{\varepsilon^2})$ rounds,

\[
\bar{x} = \frac{1}{T} (x^1 + \cdots + x^t)
\]

is an $\varepsilon$-feasible solution.

**Bounding the max number of times an element gets covered**
The Oracle

We need to implement the following oracle:

**Oracle**($\mathcal{F}, \mathcal{U}, k, p$)

\[
\sum_{S \in \mathcal{F}} x_S \leq k
\]

\[
\sum_{S \in \mathcal{F}} x_S p_S \geq 1 - \varepsilon
\]

\[
x_S \geq 0 \quad \forall S \in \mathcal{F}
\]

\[
-\phi \leq \sum_{S : e \in S} x_S - 1 \leq \phi \quad \forall e \in \mathcal{U}
\]

Given a probability vector $p$ on the elements, pick (fractionally) $k$ sets such that

1. *The total probability (weight) of the sets in the solution is maximized*, i.e., at least $(1 - \varepsilon)$, where
   - probability of a set is the sum of the probability of its elements.
2. The total number of times any *element is covered* is small.
Outline

Implementing (Streaming) MWU Oracle I
  • Naïve Solution: Heaviest Set

Implementing (Streaming) MWU Oracle II

Implementing (Streaming) MWU Oracle III
Implementing MWU in Stream (I)

**Oracle**($\mathcal{F}, \mathcal{U}, k, p$)

\[
\sum_{S \in \mathcal{F}} x_S \leq k
\]
\[
\sum_{S \in \mathcal{F}} x_S p_S \geq 1 - \varepsilon
\]
\[
x_S \geq 0 \quad \forall S \in \mathcal{F}
\]
\[
-\phi \leq \sum_{S:e \in S} x_S - 1 \leq \phi \quad \forall e \in \mathcal{U}
\]

Naïve Solution

\[x_S = \begin{cases} 
  k & \text{If } S \text{ is the heaviest set,} \\
  0 & \text{Otherwise.}
\end{cases}
\]

Width is $k$

The number of required rounds to obtain $(1 + \varepsilon)$-approximation:

\[O\left(\frac{k \log n}{\varepsilon^2}\right)\]

Find the heaviest set w.r.t $p$ in a single pass over the stream

(1 + $\varepsilon$)-approximation

$O\left(\frac{k \log n}{\varepsilon^2}\right)$ passes

$\tilde{O}(n)$ space

Performance
Implementing (Streaming) MWU Oracle I
  • Naïve Solution

Implementing (Streaming) MWU Oracle II
  • Width Reduction

Implementing (Streaming) MWU Oracle III

Performance
(1 + 𝜖)-approximation
$O\left(\frac{k \log n}{\epsilon^2}\right)$ passes
$\tilde{O}(n)$ space
Is it possible to find a solution to the oracle in set system \((U, F)\) with smaller width?

No, simply all sets may contain a designated element \(e\) and hence the width of any solution to the oracle is always \(k\) no matter how the solution is picked.

The size of an optimal cover in both set systems is the same.
Width Reduction

Is it possible to find a solution to the oracle in set system \((\mathcal{U}, \mathcal{F})\) with smaller width?

No, simply all sets may contain a designated element \(e\) and hence the width of any solution to the oracle is always \(k\) no matter how the solution is picked.

The size of an optimal cover in both set systems is the same.

Different Set System?

Extended Set System of \(\mathcal{F}\):
The set system \((\mathcal{U}, \tilde{\mathcal{F}})\) (extension of \(\mathcal{F}\)) is the collection containing all subsets of sets in \(\mathcal{F}\).

\[
\mathcal{F} = \{\{1,2,3\}, \{3,4,5\}, \{2,6\}\}
\]
\[
\tilde{\mathcal{F}} = \{
\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \\
\{1,2\}, \{1,3\}, \{2,3\}, \{3,4\}, \{3,5\}, \{4,5\}, \{2,6\}, \\
\{1,2,3\}, \{3,4,5\}
\}
\]
Width Reduction

Is it possible to find a solution to the oracle in set system \((\mathcal{U}, \mathcal{F})\) with smaller width?

No, simply all sets may contain a designated element \(e\) and hence the width of any solution to the oracle is always \(k\) no matter how the solution is picked.

The size of an optimal cover in both set systems is the same

We can easily find an optimal solution with width one in the extended set system \(\tilde{\mathcal{F}}\)

Different Set System?

Extended Set System of \(\mathcal{F}\): The set system \((\mathcal{U}, \tilde{\mathcal{F}})\) (extension of \(\mathcal{F}\)) is the collection containing all subsets of sets in \(\mathcal{F}\).

\[
\mathcal{F} = \{\{1,2,3\}, \{3,4,5\}, \{2,6\}\}
\]

\[
\tilde{\mathcal{F}} = \{
\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\},
\{1,2\}, \{1,3\}, \{2,3\}, \{3,4\}, \{3,5\}, \{4,5\}, \{2,6\},
\{1,2,3\}, \{3,4,5\}\}
\]
Width Reduction

Is it possible to find a solution to the oracle in set system \((\mathcal{U}, \mathcal{F})\) with smaller width?

No, simply all sets may contain a designated element \(e\) and hence the width of any solution to the oracle is always \(k\) no matter how the solution is picked.

The size of an optimal cover in both set systems is the same

We can easily find an optimal solution with width one in the extended set system \(\tilde{\mathcal{F}}\)

Different Set System?

Extended Set System of \(\mathcal{F}\):
The set system \((\mathcal{U}, \tilde{\mathcal{F}})\) (extension of \(\mathcal{F}\)) is the collection containing all subsets of sets in \(\mathcal{F}\).

\[
\mathcal{F} = \{\{1, 2, 3\}, \{3, 4, 5\}, \{2, 6\}\} \\
\tilde{\mathcal{F}} = \{ \\
1, \{2\}, \{3\}, \{4\}, \{5\}, \{6\} \\
\{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}, \{3, 5\}, \{4, 5\}, \{2, 6\} \\
\{1, 2, 3\}, \{3, 4, 5\} \\
\}\]
Width Reduction

Is it possible to find a solution to the oracle in set system \((\mathcal{U}, \mathcal{F})\) with smaller width?

No, simply all sets may contain a designated element \(e\) and hence the width of any solution to the oracle is always \(k\) no matter how the solution is picked.

The size of an optimal cover in both set systems is the same

We can easily find an optimal solution with width one in the extended set system \(\tilde{\mathcal{F}}\)

Different Set System?

Extended Set System of \(\mathcal{F}\):
The set system \((\mathcal{U}, \tilde{\mathcal{F}})\) (extension of \(\mathcal{F}\)) is the collection containing all subsets of sets in \(\mathcal{F}\).

\[
\mathcal{F} = \{\{1,2,3\}, \{3,4,5\}, \{2,6\}\}
\]

\[
\tilde{\mathcal{F}} = \{
\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\},
\{1,2\}, \{1,3\}, \{2,3\}, \{3,4\}, \{3,5\}, \{4,5\}, \{2,6\},
\{1,2,3\}, \{3,4,5\}
\} \]
Width Reduction

Is it possible to find a solution to the oracle in set system $(\mathcal{U}, \mathcal{F})$ with smaller width?

No, simply all sets may contain a designated element $e$ and hence the width of any solution to the oracle is always $k$ no matter how the solution is picked.

**The size of an optimal cover in both set systems is the same**

We can easily find an optimal solution with width one in the extended set system $\tilde{\mathcal{F}}$

---

**Different Set System?**

Extended Set System of $\mathcal{F}$: The set system $(\mathcal{U}, \tilde{\mathcal{F}})$ (extension of $\mathcal{F}$) is the collection containing all subsets of sets in $\mathcal{F}$.

$$\mathcal{F} = \{\{1,2,3\}, \{3,4,5\}, \{2,6\}\}$$

$$\tilde{\mathcal{F}} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{1,2\}, \{1,3\}, \{2,3\}, \{3,4\}, \{3,5\}, \{4,5\}, \{2,6\}, \{1,2,3\}, \{3,4,5\}\}$$
Width Reduction

Is it possible to find a solution to the oracle in set system \((\mathcal{U}, \mathcal{F})\) with smaller width?

No, simply all sets may contain a designated element \(e\) and hence the width of any solution to the oracle is always \(k\) no matter how the solution is picked.

The size of an optimal cover in both set systems is the same. We can easily find an optimal solution with width \(\text{one}\) in the extended set system \(\tilde{\mathcal{F}}\).

Different Set System?

Extended Set System of \(\mathcal{F}\):
The set system \((\mathcal{U}, \tilde{\mathcal{F}})\) (extension of \(\mathcal{F}\)) is the collection containing all subsets of sets in \(\mathcal{F}\).

\[
\mathcal{F} = \{\{1,2,3\}, \{3,4,5\}, \{2,6\}\}
\]

\[
\tilde{\mathcal{F}} = \{1, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{1,2\}, \{1,3\}, \{2,3\}, \{3,4\}, \{3,5\}, \{4,5\}, \{2,6\}, \{1,2,3\}, \{3,4,5\}\}
\]
Outline

Implementing (Streaming) MWU Oracle I
  • Naïve Solution

Implementing (Streaming) MWU Oracle II
  • Width Reduction via Extended Set System
  • Fractional Max $k$-Cover

Implementing (Streaming) MWU Oracle III

Performance

$(1 + \varepsilon)$-approximation $O(k \log n / \varepsilon^2)$ passes
$\tilde{O}(n)$ space
Implementing MWU in Stream (II)

\textbf{Oracle}(\mathcal{F}, \mathcal{U}, k, p)

\[ \sum_{S \in \mathcal{F}} x_S \leq k \]
\[ \sum_{S \in \mathcal{F}} x_S p_S \geq 1 - \varepsilon \]
\[ x_S \geq 0 \quad \forall S \in \mathcal{F} \]
\[ -1 \leq \sum_{S : e \in S} x_S - 1 \leq 1 \quad \forall e \in \mathcal{U} \]

Extended Set System has exponentially many sets

Solve the oracle on \( \mathcal{F} \)

Analyze w.r.t \( \tilde{\mathcal{F}} \)

- Preserved under pruning
- Imply the average constraint

Pruning the cover, reduces the total weight of the cover

Maximize the coverage instead of the total weight

Solve (Weighted) Fractional Max \( k \)-Cover in one pass
Max $k$-Cover Problem

Input: a collection $\mathcal{F}$ of sets $S_1, \ldots, S_m$
Each $S \subseteq \mathcal{U} = \{1, \ldots, n\}$

Output: $k$ subsets of $\mathcal{F}$ such that:
Maximizes the total coverage; $|\bigcup_{S \in \mathcal{C}} S|$ 

Complexity:
• NP-hard
• Greedy: $(1 - \frac{1}{e})$-approximation
• One pass $(1 - \epsilon)$-approx. using $\tilde{O}(m/\epsilon^2)$ space [MV17][BEM17]

Fractional Max $k$-Cover

Max-Cover-LP($\mathcal{F}, \mathcal{U}, k$)
\[
\begin{align*}
\text{Max.} & \quad \sum_{e \in \mathcal{U}} z_e \\
\text{s.t.} & \quad \sum_{S \in \mathcal{F}} x_S \leq k \\
& \quad \sum_{S: e \in S} x_S \geq z_e \quad \forall e \in \mathcal{U} \\
& \quad x_S \geq 0 \quad \forall S \in \mathcal{F} \\
& \quad z_e \leq 1 \quad \forall e \in \mathcal{U}
\end{align*}
\]
Max k-Cover Problem

Input: a collection $\mathcal{F}$ of sets $S_1, ..., S_m$
Each $S \subseteq \mathcal{U} = \{1, ..., n\}$

Output: $k$ subsets of $\mathcal{F}$ such that:
Maximizes the total coverage; $|\bigcup_{S \in \mathcal{C}} S|$  

Complexity:
• NP-hard
• Greedy: $(1 - \frac{1}{e})$-approximation
• One pass $(1 - \varepsilon)$-approx. using $\tilde{O}(m/\varepsilon^2)$ space [MV17], [BEM17]

Fractional (Weighted) Max k-Cover

Max-Cover-LP($\mathcal{F}$, $\mathcal{U}$, $k$, $p$)

Max. $\sum_{e \in \mathcal{U}} p_e z_e$

s.t. $\sum_{S \in \mathcal{F}} x_S \leq k$
$\sum_{S : e \in S} x_S \geq z_e \quad \forall e \in \mathcal{U}$
$x_S \geq 0 \quad \forall S \in \mathcal{F}$
$z_e \leq 1 \quad \forall e \in \mathcal{U}$
Fractional Max $k$-Cover in One Pass

Idea I (Element Sampling):
Sample $\tilde{O}(k/\varepsilon^2)$ elements according to $p$. Return the best $k$-cover of the sampled elements.

1. Sample the set $\mathcal{L}^t$ of size $\tilde{O}(k/\varepsilon^2)$ from $\mathcal{U}$ according to $p^t$
2. In one pass over the stream:
   Store $\mathcal{F}^t$, the intersection of all sets in $\mathcal{F}$ with $\mathcal{L}^t$
3. Solve Max $k$-Cover($\mathcal{L}^t, \mathcal{F}^t$)
4. In one pass: Update the vector based on the pruned cover.

w.h.p. the constructed cover is a $(1 - \varepsilon)$-approximate solution of the main instance.

MWU Update Rule:
$$p_{e}^{t+1} := p_{e}^{t}(1 - O(\varepsilon) \times (\sum x_{s}^{t} - 1))$$

Required space:
$$\tilde{O}(mk/\varepsilon^2)$$
Fractional Max $k$-Cover in One Pass

**Idea I (Element Sampling):**
Sample $\tilde{O}\left(\frac{k}{\varepsilon^2}\right)$ elements according to $p$. Return the best $k$-cover of the sampled elements.

1. Sample the set $\mathcal{L}^t$ of size $\tilde{O}\left(\frac{k}{\varepsilon^2}\right)$ from $\mathcal{U}$ according to $p^t$
2. In one pass over the stream:
   Store $\mathcal{F}^t$, the intersection of all sets in $\mathcal{F}$ with $\mathcal{L}^t$
3. Solve Max $k$-Cover($\mathcal{L}^t, \mathcal{F}^t$)
4. In one pass: Update the vector based on the pruned cover.

w.h.p. the constructed cover is a $(1 - \varepsilon)$-approximate solution of the main instance.

**Idea II:** In the preprocessing step, pick $x_{cmn} = \left\{\frac{\varepsilon k}{m}, ..., \frac{\varepsilon k}{m}\right\}$

Required space: $\tilde{O}(mk/\varepsilon^2)$

Required space: $\tilde{O}\left(\frac{m}{\varepsilon k} \times \frac{k}{\varepsilon^2}\right) = \tilde{O}(m/\varepsilon^3)$

$e$ is covered
$\quad p^t_e, e$ is not covered
Implementing MWU in Stream (II)

Oracle($\mathcal{F}, \mathcal{U}, k, p$)

$\sum_{S \in \mathcal{F}} x_S \leq k$

$\sum_{S \in \mathcal{F}} x_S p_S \geq 1 - \varepsilon$

$x_S \geq 0 \quad \forall S \in \mathcal{F}$

$-1 \leq \sum_{S : e \in S} x_S - 1 \leq 1 \quad \forall e \in \mathcal{U}$

Pruning the cover, reduces the total weight of the cover

Maximize the coverage instead of the total weight

Solve (Weighted) Fractional Max $k$-Cover in one pass

The number of required rounds to obtain $(1 + \varepsilon)$-approximation:

$O\left(\frac{\log n}{\varepsilon^2}\right)$

Find max $k$-cover w.r.t $p$ in two passes over the stream using $\tilde{O}\left(\frac{m}{\varepsilon^3}\right)$ space

(1 + $\varepsilon$)-approximation

$O\left(\frac{\log n}{\varepsilon^2}\right)$ passes

$\tilde{O}\left(\frac{m}{\varepsilon^3}\right)$ space

Performance

Width is 1
Outline

Implementing (Streaming) MWU Oracle I
  • Naïve Solution

Implementing (Streaming) MWU Oracle II
  • Reducing Width via Extended Set System
  • Fractional Max $k$-Cover

Implementing (Streaming) MWU Oracle III
  • Sampling in Advance
Reducing the Number of Passes Further!

Perform several rounds of MWU in one pass
	× But probability distribution \( p \) changes over the iterations
	× Element sampling is done w.r.t. \( p \)

Key observation:
The probability vector \( p \) changes slowly.

After \( \ell \) rounds of MWU:
\[
p_{e+\ell}^t \leq p_e^t (1 + O(\varepsilon))^\ell
\]

Setting \( \ell = O\left(\frac{\log n}{\varepsilon^2 d}\right) \) rounds, \( p_e \)
increases at most by \( n^{O\left(\frac{1}{\varepsilon d}\right)} \)

Idea I (Element Sampling):
Sample \( \tilde{O}\left(\frac{k}{\varepsilon^2}\right) \) elements according to \( p \).
Return the best \( k \)-cover of the sampled elements.
Reducing the Number of Passes Further!

Perform several rounds of MWU in one pass
- But probability distribution $p$ changes over the iterations
- Element sampling is done w.r.t. $p$

Key observation:
The probability vector $p$ changes slowly.

After $\ell$ rounds of MWU:
$$p_{e}^{t+\ell} \leq p_{e}^{t}(1+O(\varepsilon))^{\ell}$$

Setting $\ell = O\left(\frac{\log n}{\varepsilon^2 d}\right)$ rounds, $p_{e}$ increases at most by $n^{O\left(\frac{1}{\varepsilon d}\right)}$

Idea I (Element Sampling):
Sample $\tilde{O}\left(\frac{kn^{O(1/\varepsilon d)}}{\varepsilon^2}\right)$ elements according to $p$.
Return the best $k$-cover of the sampled elements.

Rejection Sampling: To adjust the probability $p_{e}$
Keep each sample w.p. $p_{e}^{t+\ell}/p_{e}^{t}n^{O(1/\varepsilon d)}$

To perform $O\left(\frac{\log n}{\varepsilon^2 d}\right)$ rounds together
Reducing the Number of Passes Further!

Perform several rounds of MWU in one pass
  - But probability distribution $p$ changes over the iterations
  - Element sampling is done w.r.t. $p$

Key observation:
The probability vector $p$ changes slowly.

After $\ell$ rounds of MWU:
$$p_e^{t+\ell} \leq p_e^t (1 + O(\epsilon))^\ell$$

Setting $\ell = O\left(\frac{\log n}{\epsilon^2 d}\right)$ rounds, $p_e$ increases at most by $n^{O\left(\frac{1}{\epsilon d}\right)}$

Idea I (Element Sampling):
Sample $\tilde{O}\left(\frac{kn^{O(1/\epsilon d)}}{\epsilon^2}\right)$ elements according to $p$.
Return the best $k$-cover of the sampled elements.

Rejection Sampling: To adjust the probability $p_e$

Space increases by $n^{O(1/\epsilon d)}$
#passes decreases by $O\left(\frac{\log n}{\epsilon^2 d}\right)$
Implementing MWU in Stream (III)

**Oracle**$(\mathcal{F}, \mathcal{U}, k, p)$

\[
\begin{align*}
\sum_{S \in \mathcal{F}} x_S &\leq k \\
\sum_{S \in \mathcal{F}} x_S p_S &\geq 1 - \varepsilon \\
\end{align*}
\]

\[x_S \geq 0 \quad \forall S \in \mathcal{F}\]

\[-\phi \leq \sum_{S: e \in S} x_S - 1 \leq \phi \quad \forall e \in \mathcal{U}\]

Width is 1

The number of required rounds to obtain $(1 + \varepsilon)$-approximation: \(O\left(\frac{\log n}{\varepsilon^2}\right)\)

Pruning the cover, reduces the total weight of the cover

Maximize the coverage instead of the total weight

Solve *(Weighted) Fractional Max \(k\)-Cover* in one pass

Sampling in advance for \(O\left(\frac{\log n}{\varepsilon^2 d}\right)\) rounds.

(1 + \(\varepsilon\))-approximation \(O(d)\) passes \(\widetilde{O}(mn^{O(1/d\varepsilon)})\) space

Find max \(k\)-cover w.r.t \(p\) in two passes over the stream using \(\widetilde{O}(mn^{O(1/d\varepsilon)})\) space

Performance
Sampling in advance for $O\left(\frac{\log n}{\varepsilon^2 d}\right)$ rounds.

The number of required rounds to obtain $(1 + \varepsilon)$-approximation: $O\left(\frac{\log n}{\varepsilon^2}\right)$

Find max $k$-cover w.r.t $p$ in a single pass over the stream using $\tilde{O}(mn^{O(1/d\varepsilon)})$ space.

Open Questions:

1. Better bound for general covering/packing LP?
2. Any constant pass polylog-approximation algorithm for Weighted Set Cover with $o(mn)$ space?
3. Optimal number of passes for $O(\log n)$-approx. Set Cover?
   1. Best Upper Bound: $O(\log n)$-pass
   2. Best Lower Bound: $\Omega\left(\frac{\log n}{\log \log n}\right)$-pass [CW16]

Performance

$(1 + \varepsilon)$-approximation $O(d)$ passes $\tilde{O}(mn^{O(1/d\varepsilon)})$ space

Thank You