INFO-F440 - ALGORITHMS FOR BIG DATA - 202324

Maximum Coverage in Random-Arrival Streams

Presenter:
Herma ELEZI
Nicolas BONGAY

2024

Table of Content

Introduction

- Maximum Coverage Problem.

Definition: Given a collection of m sets, each a subset of a universe $\{1,\ldots,n\}$, select k sets whose union has the largest cardinality.

- Goal: Choose k sets from a collection such that their union has the largest cardinality.

Maximum Coverage

ILP formulation:

$$
\begin{array}{lll}
\text{maximize} & \sum_{e_j \in E} y_j & \text{(maximizing the sum of covered elements)} \\
\text{subject to} & \sum x_i \leq k & \text{(no more than } k \text{ sets are selected)} \\
 & \sum_{e_j \in S_i} x_i \geq y_j & \text{(if } y_j > 0 \text{ then at least one set } e_j \in S_i \text{ is selected)} \\
 & y_j \in \{0,1\} & \text{(if } y_j = 1 \text{ then } e_j \text{ is covered)} \\
 & x_i \in \{0,1\} & \text{(if } x_i = 1 \text{ then } S_i \text{ is selected for the cover)}
\end{array}
$$

NP-Hard

(Greedy algorithm:
1-1/e approximation)

Streaming Models

Set-Streaming Model

Edge-Streaming Model

Random-Arrival Model:

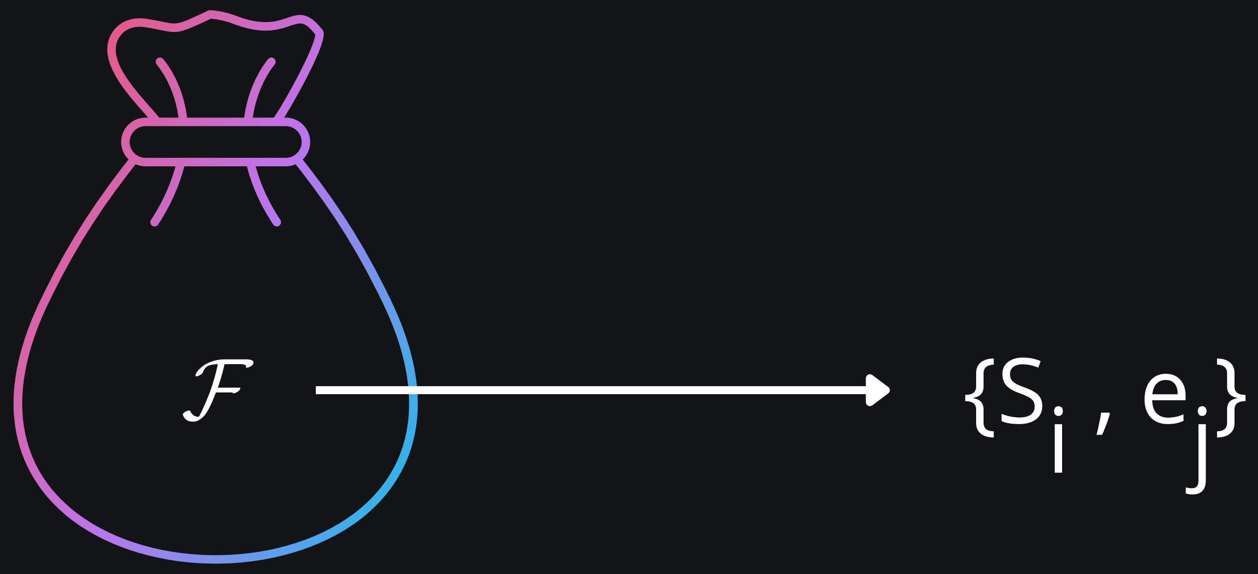
Arbitrary-Arrival Model:

Data Stream Processing

Set-streaming model:

each set is contiguously listed

Edge-streaming model:



Each random selection
gives a pair (Set i, element j)

What we know:

Set-streaming:

Arbitrary-arrival: 1/2 approximation is the best (low space)

Random-arrival: best approximation is between 1/2 and 1-1/e (low space)

Breaking the 1-1/e (≈0.63) approximation requires high space

The best approximation at that moment is 1/2

Edge-streaming:

α-approximation in $\tilde{O}(\alpha^2 m)$ space

high space

The objective:

By mixing algorithms from many different papers, they aim to:

- Solve Set-streaming random-arrival model
- WIth a One-pass algorithm
- Using low space (polylogarithmic in n,m)
- With more than a 1/2 approximation

The main idea is to combine two algorithms to perform the maximum coverage

MV-4 for the sumbsampling of the data

ANY subroutine that computes the submodular maximization problem (e.g. SALSA)

MV-4

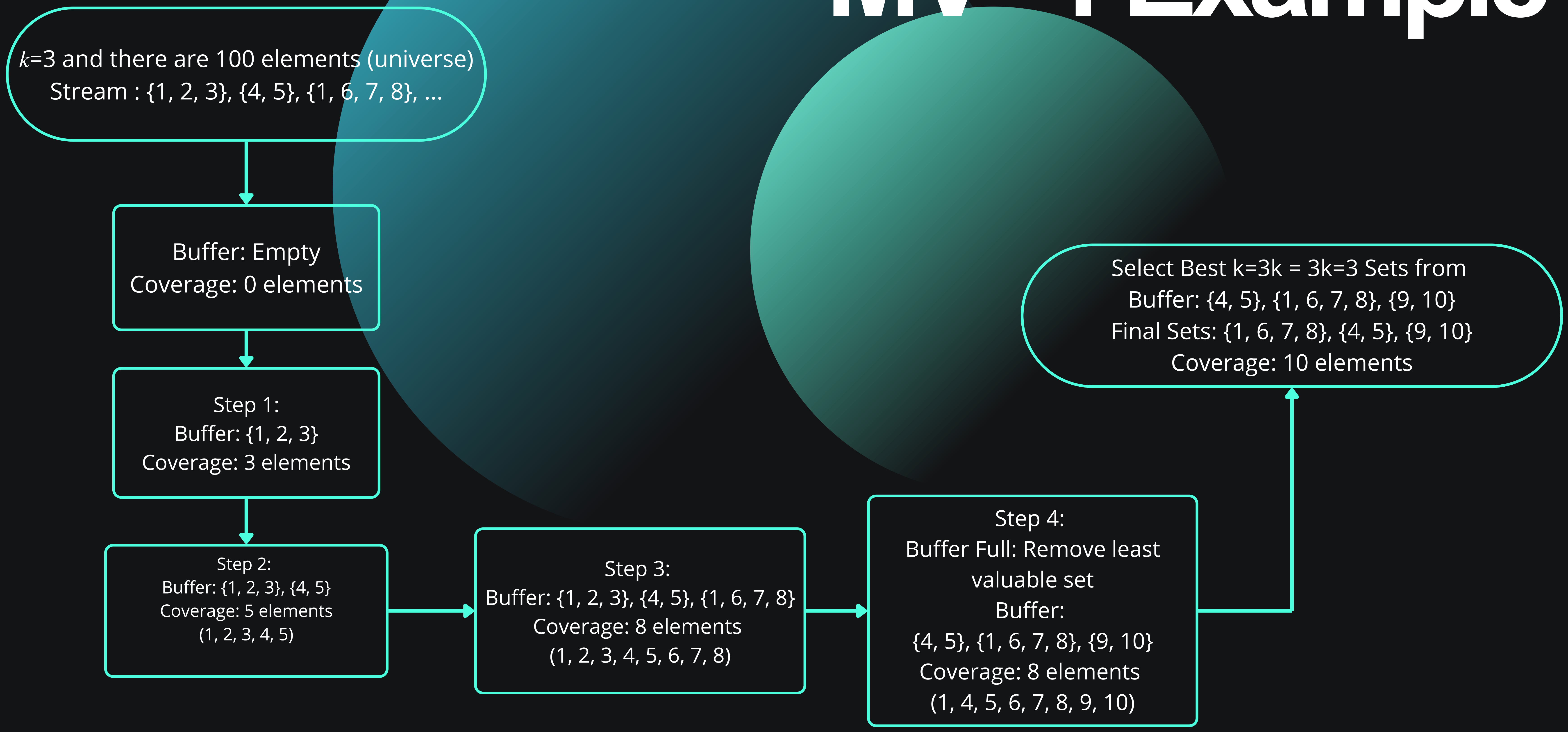
MV-4 is a set-streaming maximum coverage algorithm designed to operate in an arbitrary-arrival model.

Approximation factor $\longrightarrow$ $\frac{1}{2} - \varepsilon$ using $\tilde{O}(k\varepsilon^{-3})$ space

How MV- 4 Works:

1. Initialization
2. Processing Stream
3. Selection Criteria
4. Final Selection

MV- 4 Example



k=3 and there are 100 elements (universe)
Stream : {1, 2, 3}, {4, 5}, {1, 6, 7, 8}, ...
Buffer: Empty
Coverage: 0 elements
Step 1:
Buffer: {1, 2, 3}
Coverage: 3 elements
Step 2:
Buffer: {1, 2, 3}, {4, 5}
Coverage: 5 elements
(1, 2, 3, 4, 5)
Step 3:
Buffer: {1, 2, 3}, {4, 5}, {1, 6, 7, 8}
Coverage: 8 elements
(1, 2, 3, 4, 5, 6, 7, 8)
Step 4:
Buffer Full: Remove least valuable set
Buffer:
{4, 5}, {1, 6, 7, 8}, {9, 10}
Coverage: 8 elements
(1, 4, 5, 6, 7, 8, 9, 10)
Select Best k=3k = 3k=3 Sets from
Buffer: {4, 5}, {1, 6, 7, 8}, {9, 10}
Final Sets: {1, 6, 7, 8}, {4, 5}, {9, 10}
Coverage: 10 elements

SALSA

SALSA (Streaming Algorithm for Large Set Approximation) algorithm is to provide a near-optimal solution to the maximum coverage problem in a streaming context with random arrival orders.

Approximation factor $\longrightarrow$ $1 - \frac{1}{2} + co$

How SALSA Works:

01

Initialization

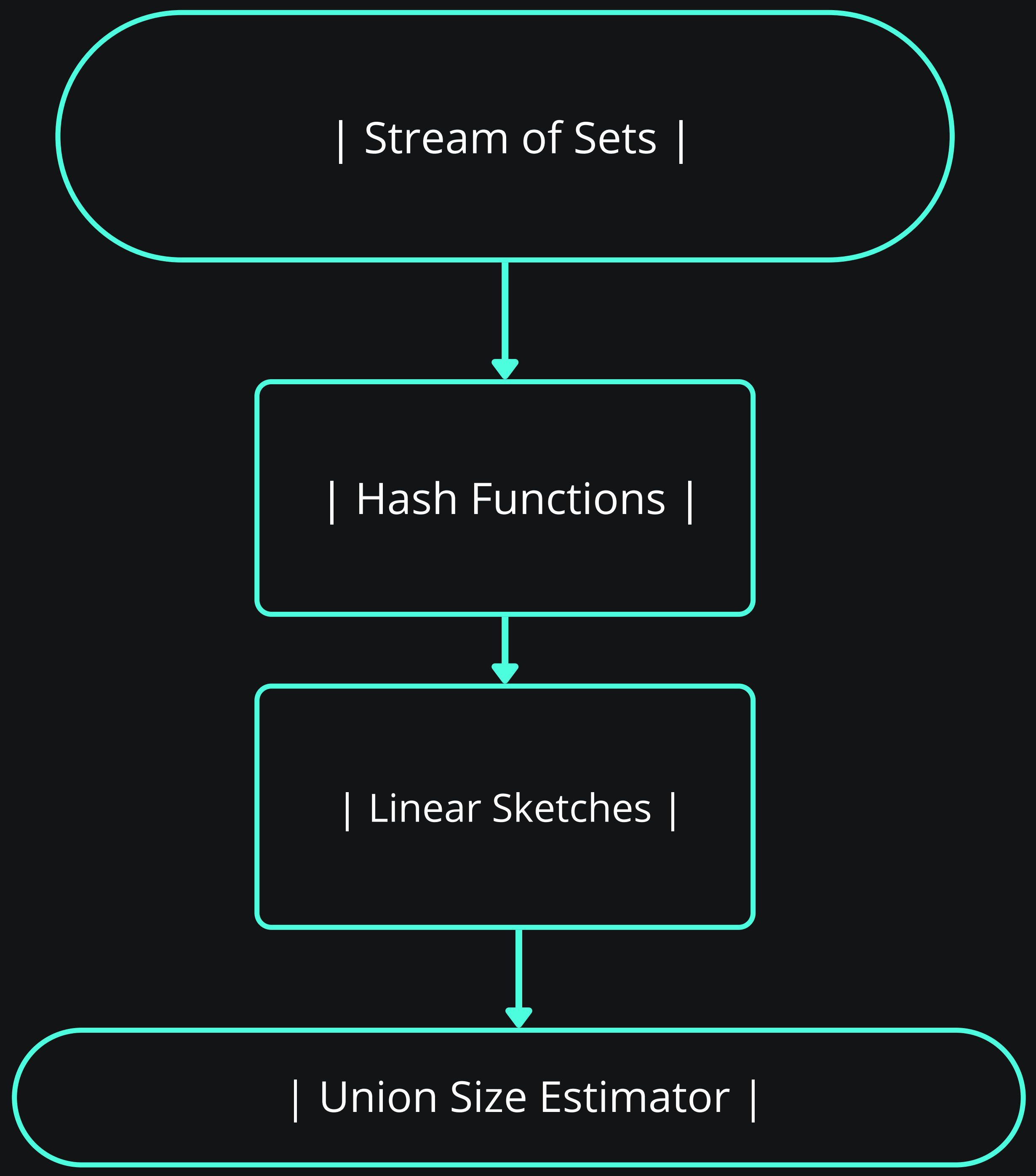
02

Processing Stream

03

Final Selection

SALSA Example:



Assumptions:

There is w such that : d/2 ≤ w ≤ d (d≤OPT≤kd)

There exists a family of λ-wise independant binary hash functions such that Pr(h(e)=1)=p_w

An instantiation of B shouldn't use more than a certain amount of space, else it gets terminated

▶ **Lemma 3.** *With probability at least $1-\varepsilon$, for all collections of up to k sets $S_1, \ldots, S_l \in \mathcal{F}$, $|S'_1 \cup \cdots \cup S'_l| = p \cdot |S_1 \cup \cdots \cup S_l| \pm p\varepsilon d$.*

▶ **Corollary 5.** *Let OPT' be the optimal coverage for the subsampled problem instance $\mathcal{I}'$. If a choice of k sets $S_1, \ldots, S_k$ satisfies $|S'_1 \cup \cdots \cup S'_k| \geq \beta \cdot \mathrm{OPT}'$, then with probability at least $1-\varepsilon$, $|S_1 \cup \cdots \cup S_k| \geq (\beta - 2\varepsilon) \cdot \mathrm{OPT}$.*

Generalised Subsampling Algorithm GS(B)

Algorithm 1 The generalised subsampling algorithm GS($\mathcal{B}$).

$W \leftarrow \{2^i : i \in \mathbb{N},\ 2^i \leq n\}$ ▷ Guesses for d
$\lambda \leftarrow \lfloor 2k \log(em\varepsilon^{-1}) \rfloor$ ▷ Hash function independence parameter
for $w \in W$ **do** O(log(n))
  Initialise $\mathcal{B}_w$, an instantiation of $\mathcal{B}$ SALSA
  $p_w \leftarrow \min\{1, 3k\varepsilon^{-2}\log(em\varepsilon^{-1})w^{-1}\}$ ▷ Subsampling rate
  Sample $h_w \in \mathcal{H}_{p_w,\lambda}$ uniformly at random ▷ Choose hash function
  $\texttt{active}_w \leftarrow \texttt{true}$ ▷ Kill switch indicator

MV-4
for $i = 1, \ldots, m$ **do** O(m) ▷ Iterate over each set in the stream
  for $e \in S_i$ **do** O(d) ▷ Iterate over each element in the set
    for $w \in W$ **do** O(log(n))
      if $d^*_w > 2p_w w(1+\varepsilon)$ **then** Too much space ?
        $\texttt{active}_w \leftarrow \texttt{false}$ ▷ Terminate $\mathcal{B}_w$
      if $\texttt{active}_w$ and $h_w(e) = 1$ **then**
        Supply e to the stream of $\mathcal{B}_w$
Let $I_w \subset [m]$ be the solution returned by $\mathcal{B}_w$
$w_c \leftarrow \min\{w \in W : d^*/2 \leq w \leq d^*\}$ ▷ At this point, $d^* = d$.
return I_{w_c}

GS-SALSA

▶ **Theorem 2** (Generalised subsampling). *Suppose $\mathcal{B}$ achieves an approximation factor of α in-expectation*[9] *for set-streaming maximum coverage using $O(ds)$ space, where d is the maximum set size. There exists an algorithm, called GS($\mathcal{B}$), which, given $\varepsilon > 0$, achieves an approximation factor of $\alpha - \varepsilon$ in-expectation and uses $\tilde{O}(k\varepsilon^{-2}s)$ space.*

Space used: $\tilde{O}(k^2)$

Running time: $\tilde{O}(T_B dm)$

1/2 + c_1 approximation

k: # of sets to select
d: size of bigest set
m: # of sets
T_B : Running time of B

Here, Salsa runs in a polylogarthmic time

What about Edge-streaming ?

They prooved that $\forall \alpha > 0$, $\forall \delta > 0$:

Any algorithm that α-approximates with proba at least 1-δ requires $\Omega(m^{1-\delta})$ space

So, no low space algorithm

Previous Work

MV-4 : $1/2 - \varepsilon$ approximation

(Andrew McGregor and Hoa T Vu. Better streaming algorithms for the maximum coverage problem. Theory of Computing Systems, 63:1595–1619, 2019)

SALSA: $1/2 + c_0$

(Ashkan Norouzi-Fard, Jakub Tarnawski, Slobodan Mitrovic, Amir Zandieh, Aidasadat Mousavifar, and Ola Svensson. Beyond 1/2-approximation for submodular maximization on massive
data streams. In 35th ICML, pages 3829–3838, 2018.)

SMC: $1 - 1/e - \varepsilon - o(1)$

(Shipra Agrawal, Mohammad Shadravan, and Cliff Stein. Submodular Secretary Problem with Shortlists. In 10th ITCS, pages 1:1–1:19, 2019.)

Greedy algorithm : 1-1/e (But high space)