Submodular Streaming in All Its Glory:
Tight Approximation, Minimum Memory and Low Adaptive Complexity

Ehsan Kazemi[1], Marko Mitrovic[1], Morteza Zadimoghaddam[2], Silvio Lattanzi[2], Amin Karbasi[1]

[1]Yale University and [2]Google

Streaming Algorithms

- Many practical scenarios we need to use streaming algorithms:
  - the data arrives at a very fast pace
  - there is only time to read the data once
  - random access to the entire data is not possible and only a small fraction of the data can be loaded to the main memory

Video from “Britain’s Got Talent”

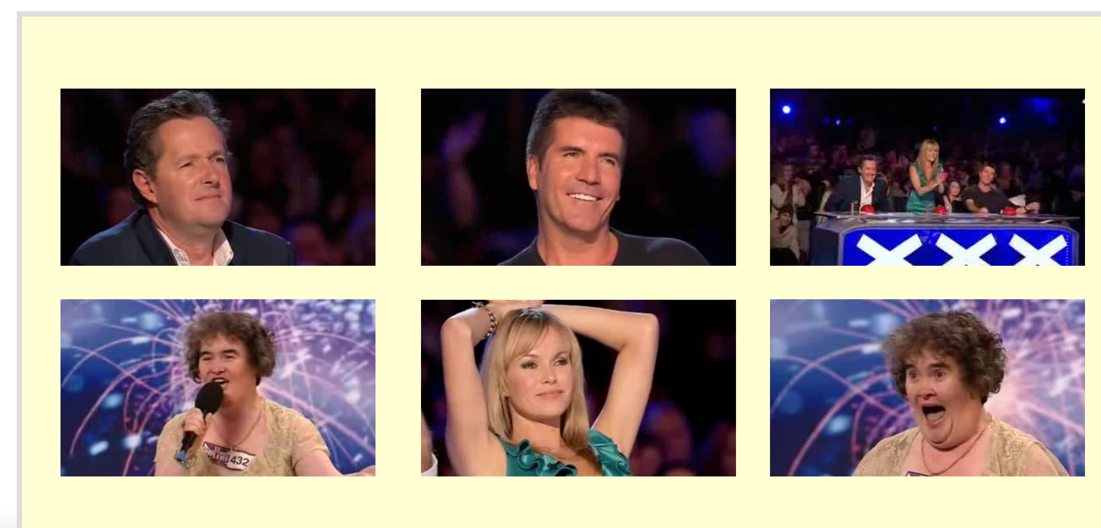
Summary

Streaming Algorithms

- Many practical scenarios we need to use streaming algorithms:
  - the data arrives at a very fast pace
  - there is only time to read the data once

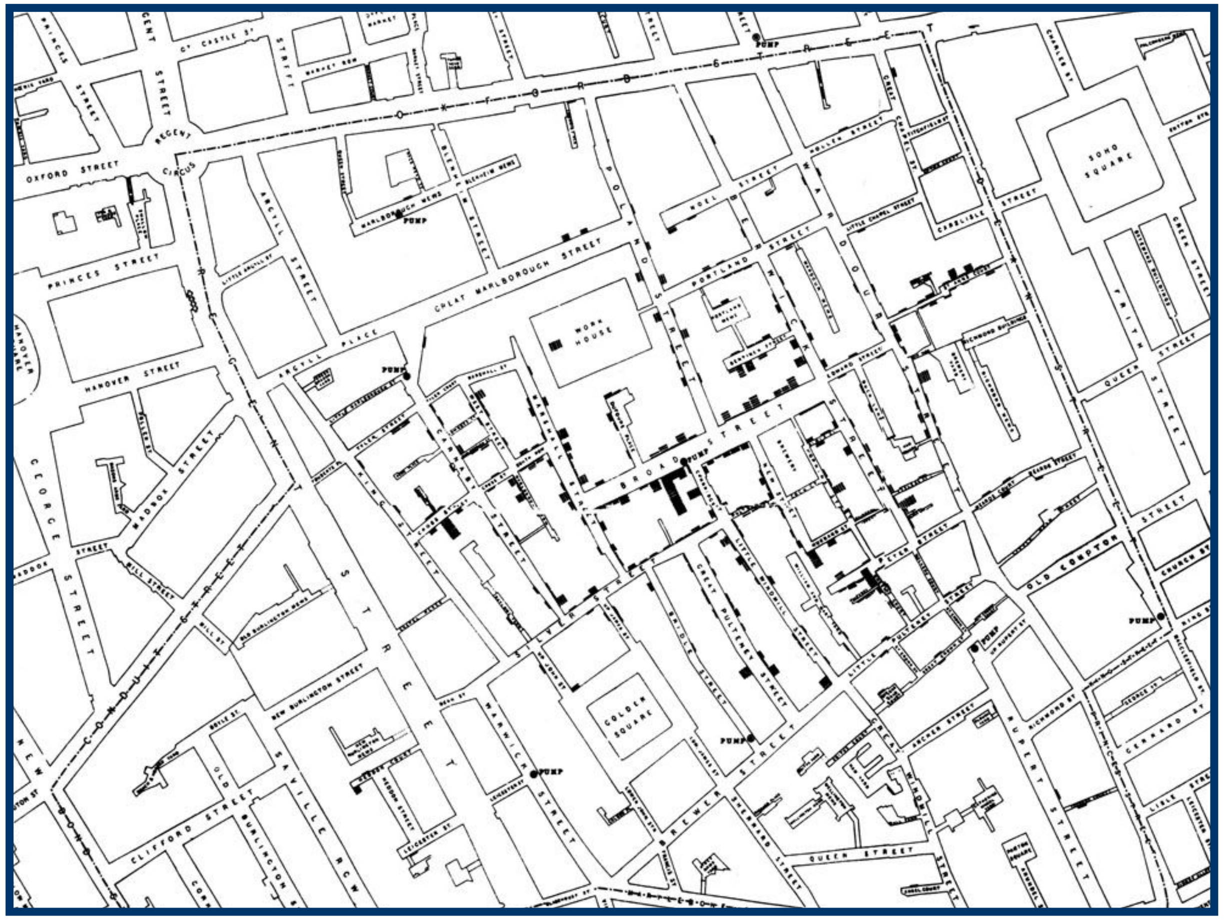
Is it possible to summarize a massive data set "on the fly", i.e., when at any point of time we have access only to a small fraction of data?

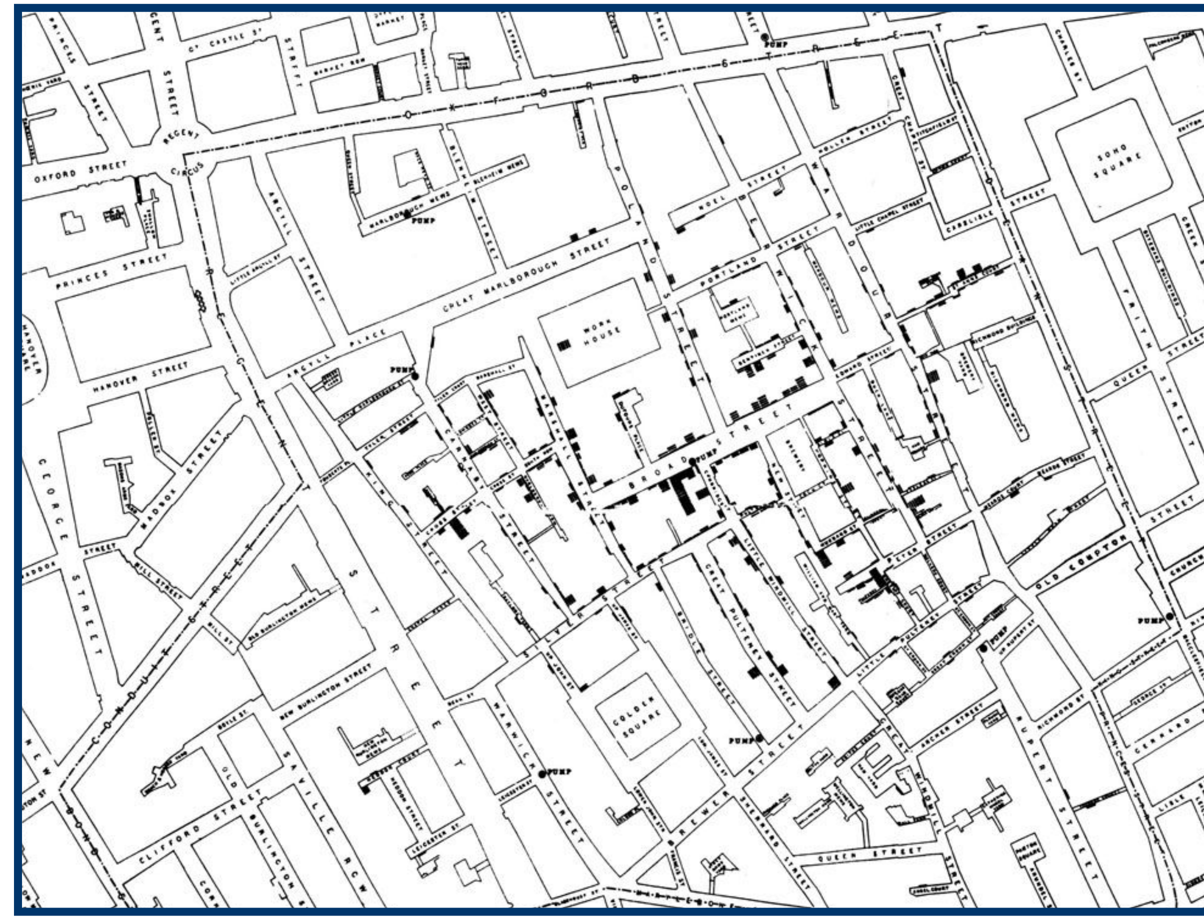
Video from "Britain's Got Talent"



Summary

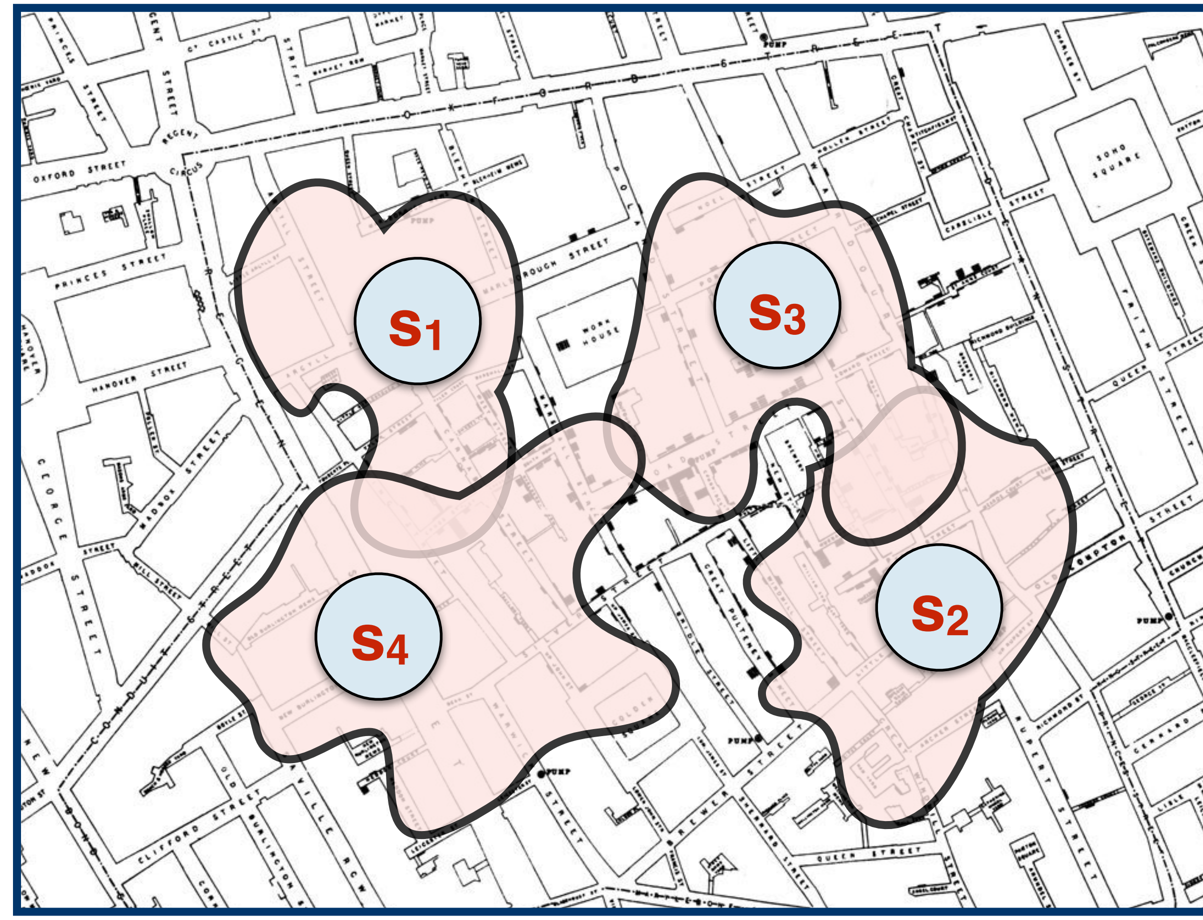
Submodularity





Submodularity

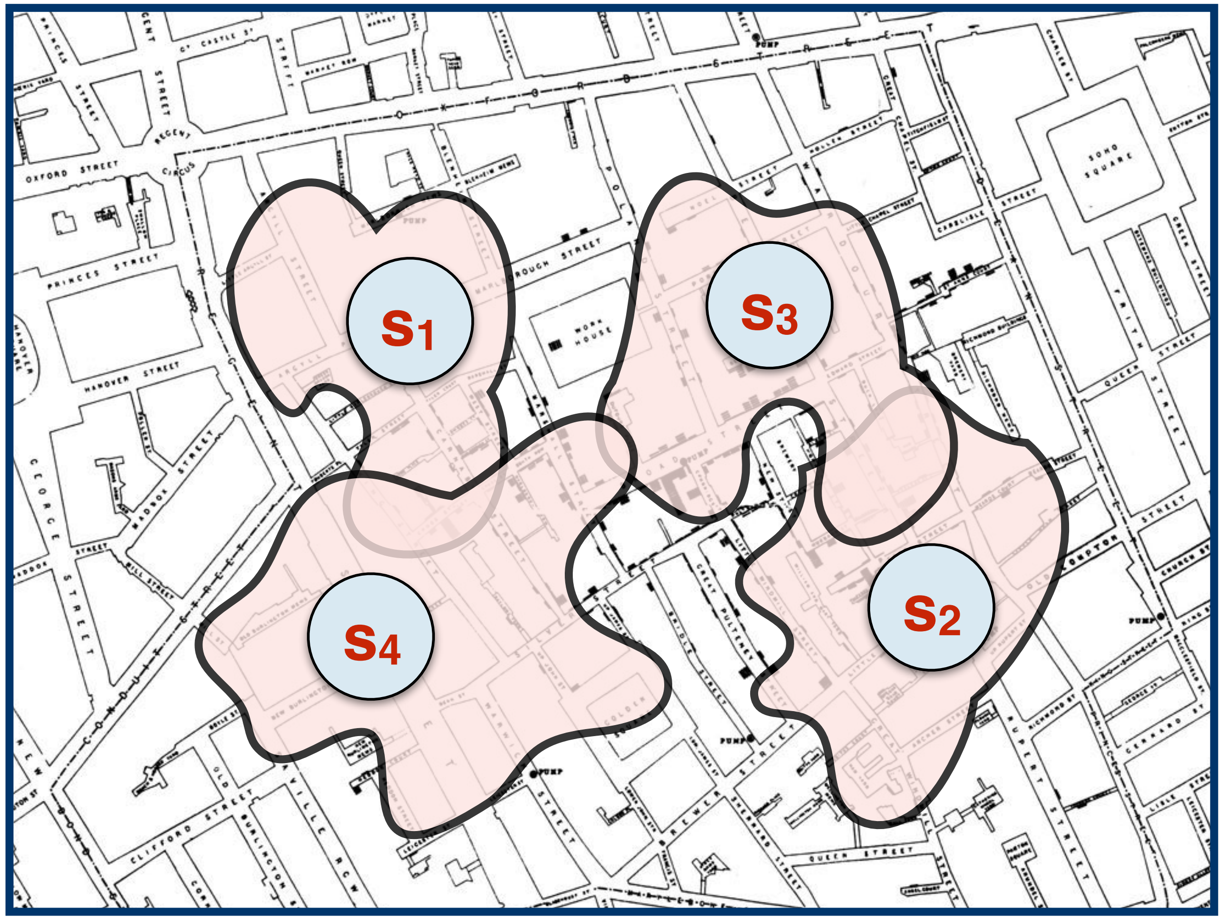
$A = \{s_1, s_2\}$



$B = \{s_1, s_2, s_3, s_4\}$

Submodularity

$A = \{s_1, s_2\}$

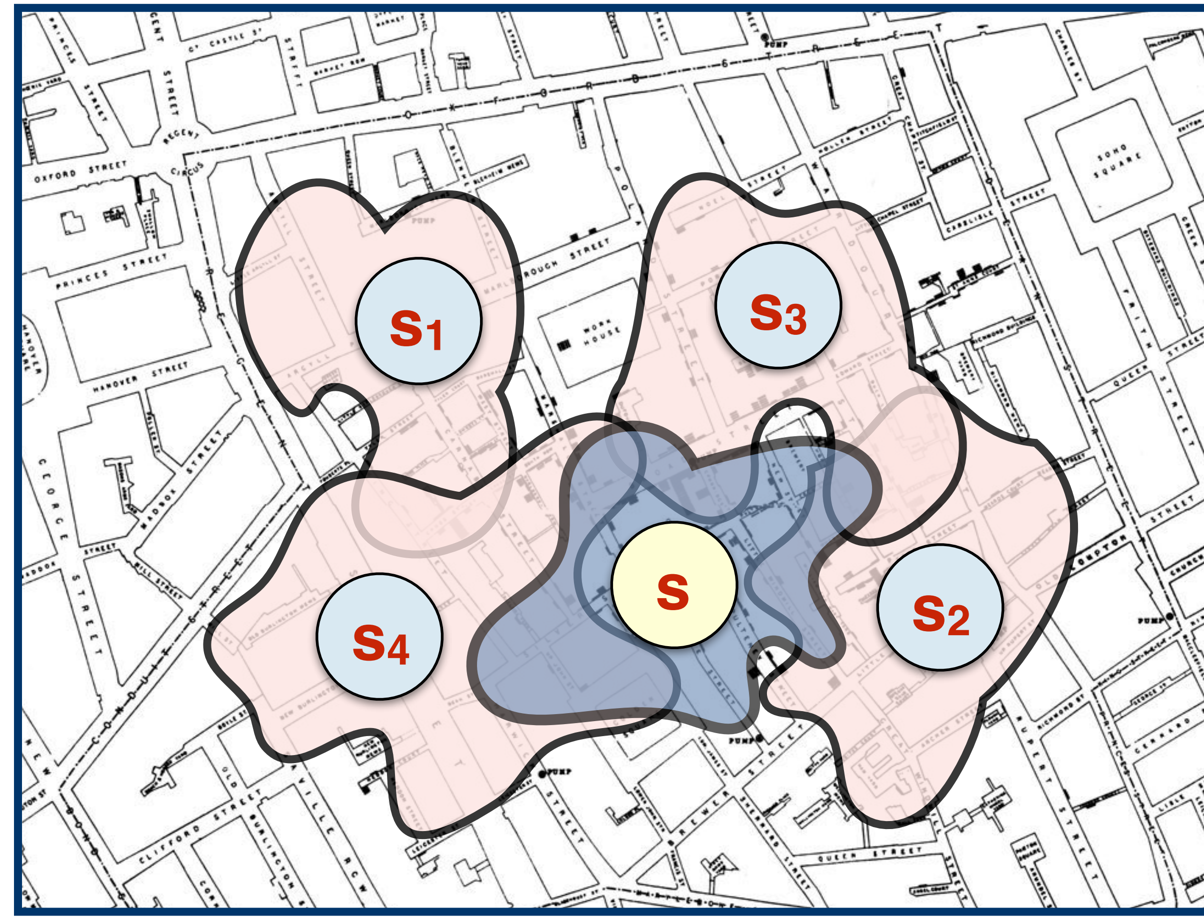


$B = \{s_1, s_2, s_3, s_4\}$

Submodularity

Submodularity

$$A = \{s_1, s_2\}$$

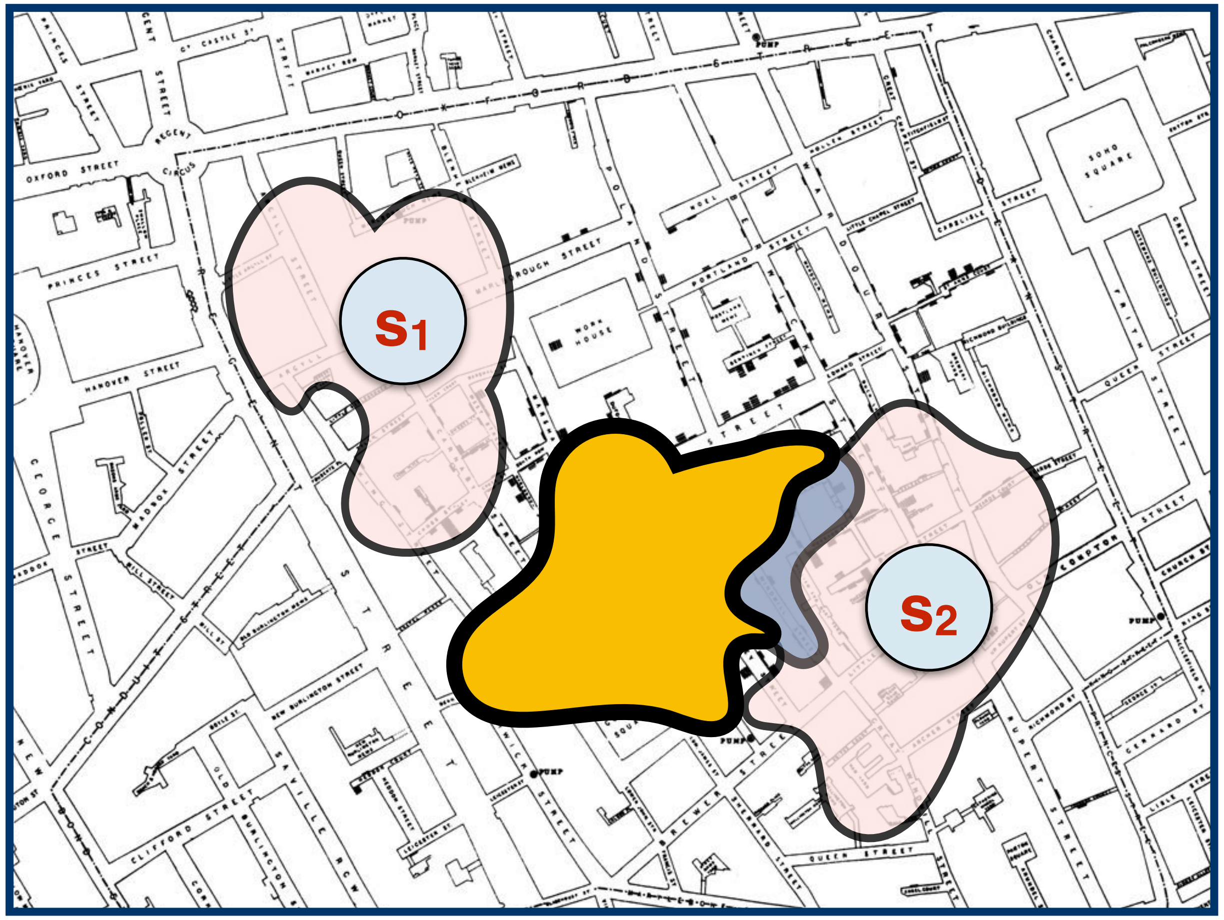


$$B = \{s_1, s_2, s_3, s_4\}$$

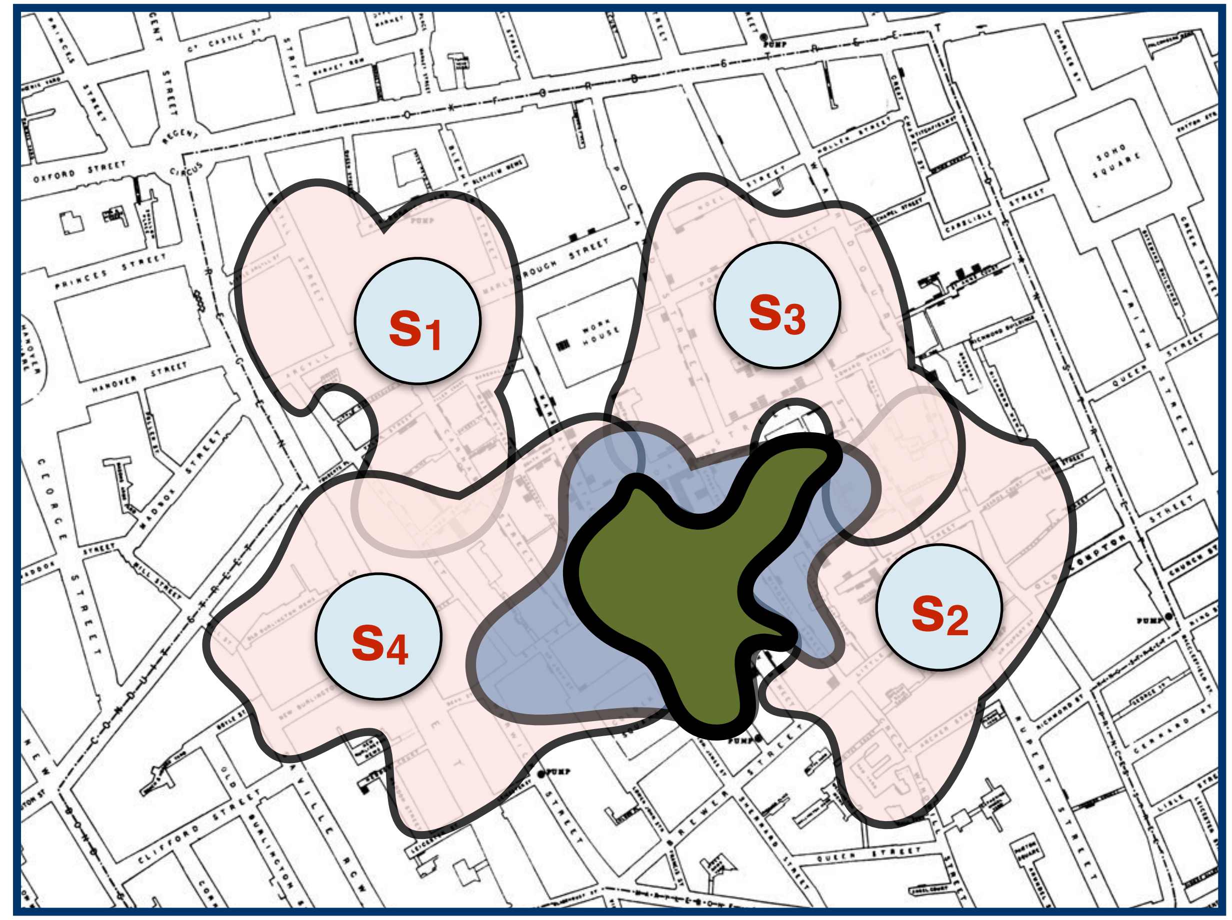
Submodularity

$$\forall\ A \subseteq B \subseteq V \text{ and } s \in V \setminus B$$

Submodularity



$$A = \{s_1, s_2\}$$

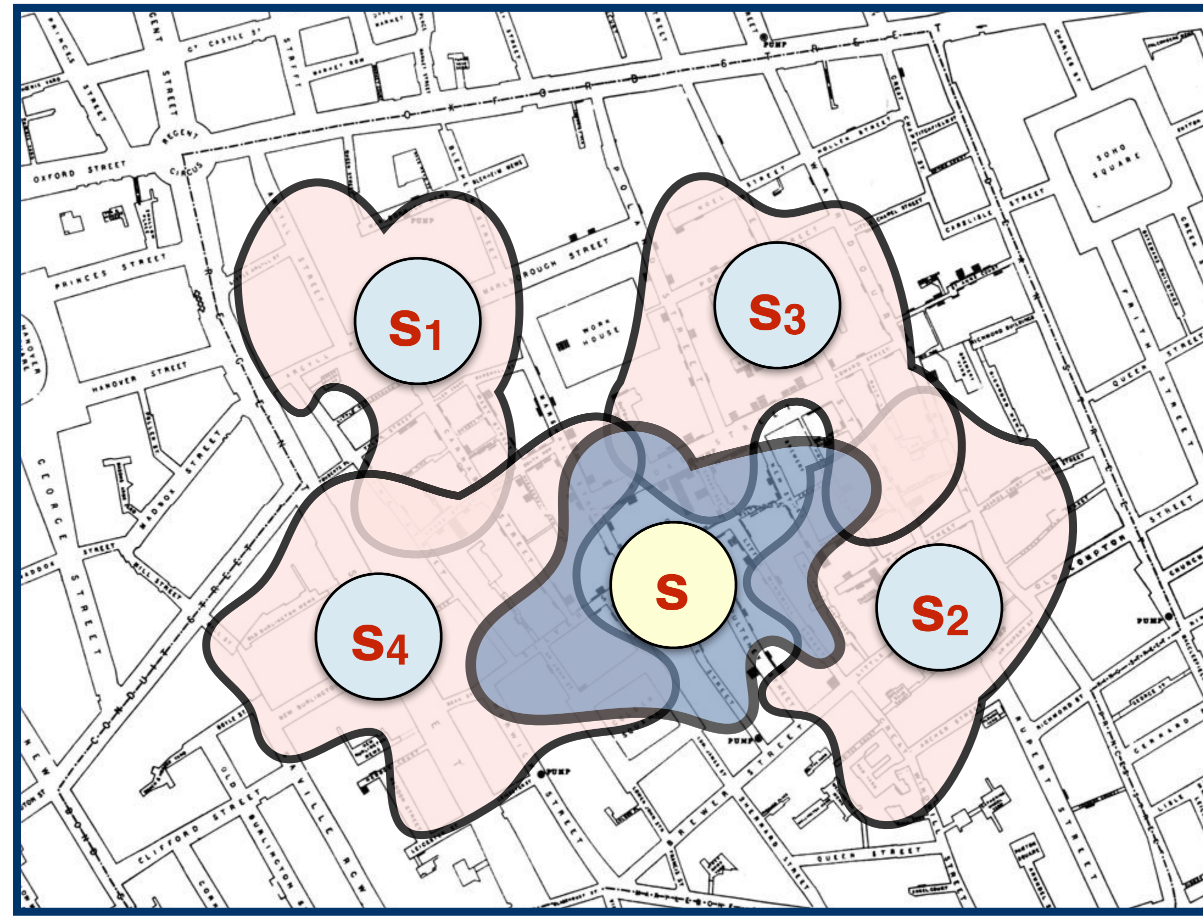


$$B = \{s_1, s_2, s_3, s_4\}$$

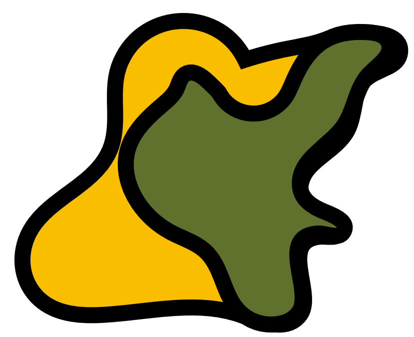
Submodularity

$$\forall\ A \subseteq B \subseteq V \text{ and } s \in V \setminus B$$

Submodularity



$$A = \{s_1, s_2\}$$



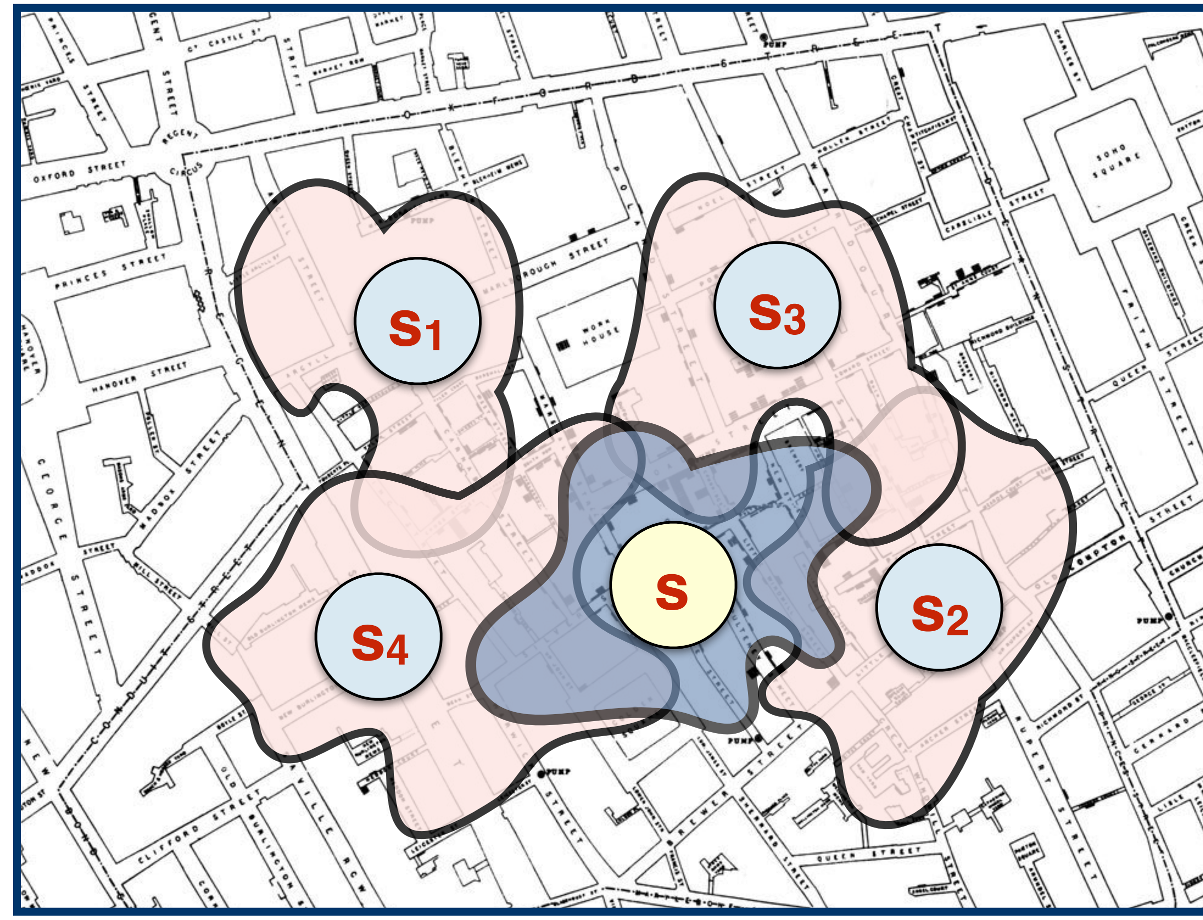
$$B = \{s_1, s_2, s_3, s_4\}$$

Submodularity

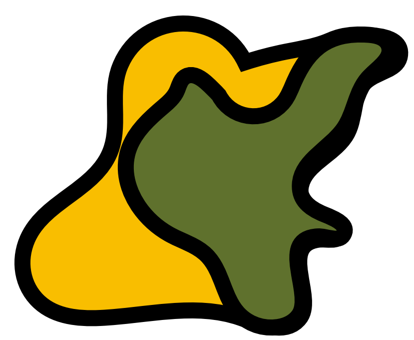
$$\forall\ A \subseteq B \subseteq V \text{ and } s \in V \setminus B$$

$$f(A \cup \{s\}) - f(A) \geq f(B \cup \{s\}) - f(B)$$

Submodularity



$A = \{s_1, s_2\}$



$B = \{s_1, s_2, s_3, s_4\}$

Submodularity

$\forall\ A \subseteq B \subseteq V$ and $s \in V \setminus B$

$f(A \cup \{s\}) - f(A) \geq f(B \cup \{s\}) - f(B)$

Monotonicity

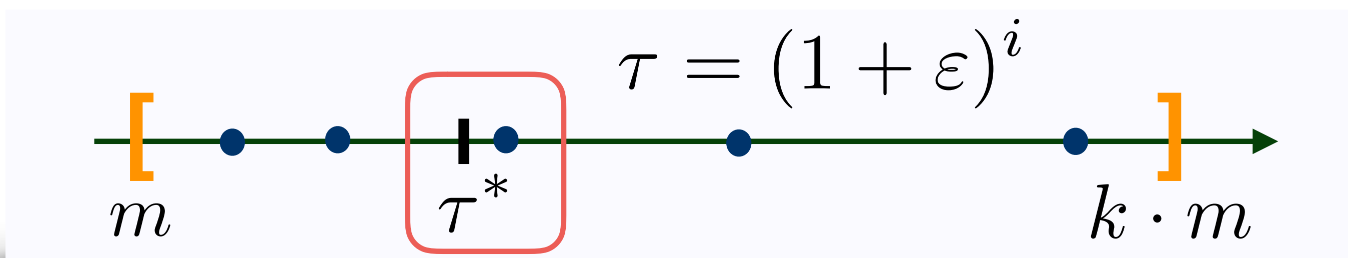
$\forall A \subseteq B \subseteq V$

$f(A) \leq f(B)$

Yale

Closing the Gap: SIEVE-STREAMING++

- Choosing elements with marginal gain at least $\tau^* = \mathrm{OPT}/(2k)$ returns a set S with an objective value of at least $f(S) \geq \mathrm{OPT}/2$
- Can we find a good approximation of OPT?
  - Submodularity: $m \leq \mathrm{OPT} \leq k \cdot m, m = \max_{e \in V} f(\{e\})$
  - Find better lower bounds for OPT as more elements arrive



[Kazemi, Mitrovic, Zadimoghaddam, Lattanzi, Karbasi]

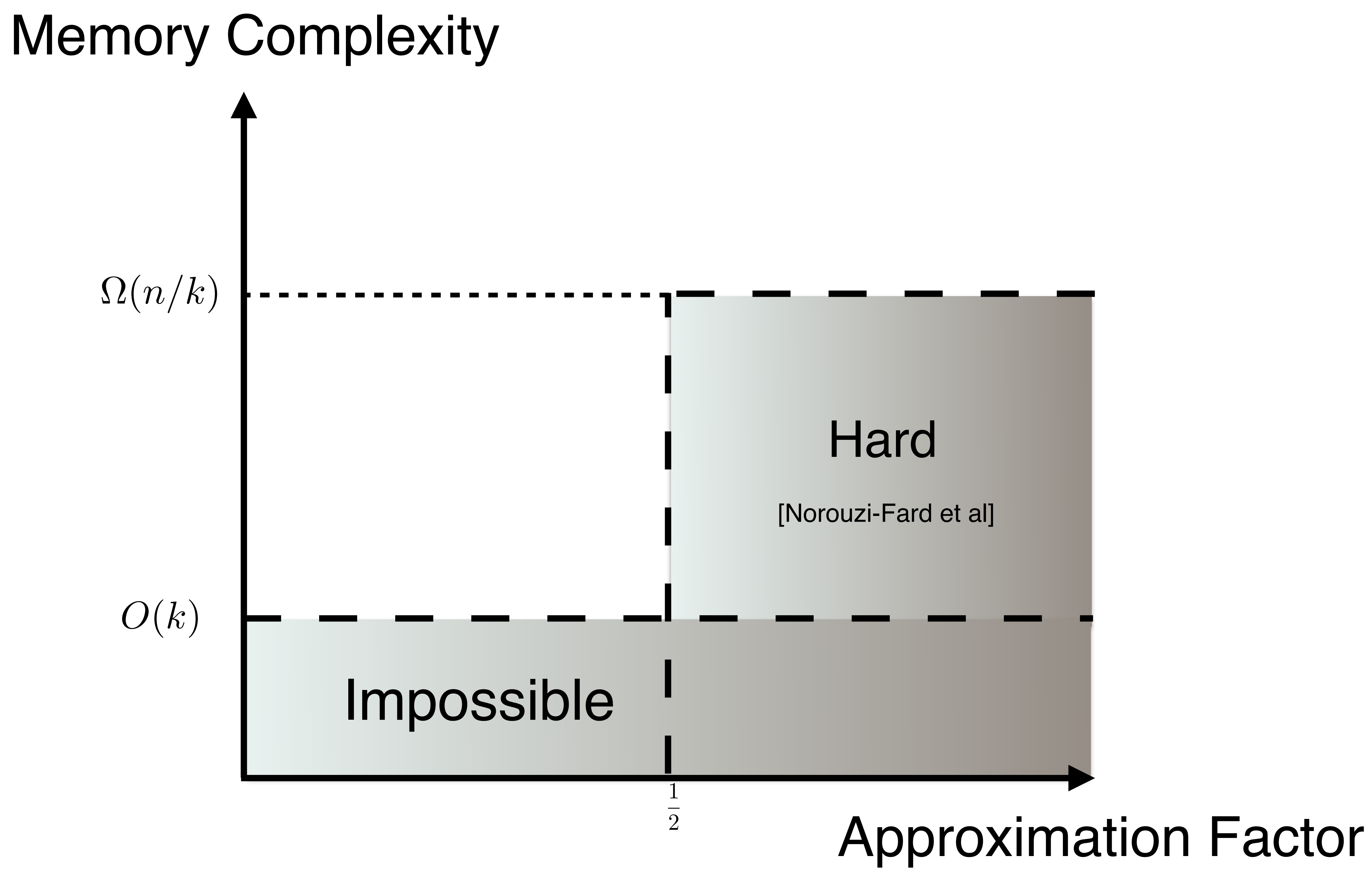
For monotone submodular functions, SIEVE-STREAMING++ with a $O(k)$ memory gives constant factor approximation using only $O(\log(k)/\varepsilon)$ function evaluation per element

$$f\left(S_{\text{SIEVE-STREAMING++}}\right) \geq (1/2)\ \mathrm{OPT}$$

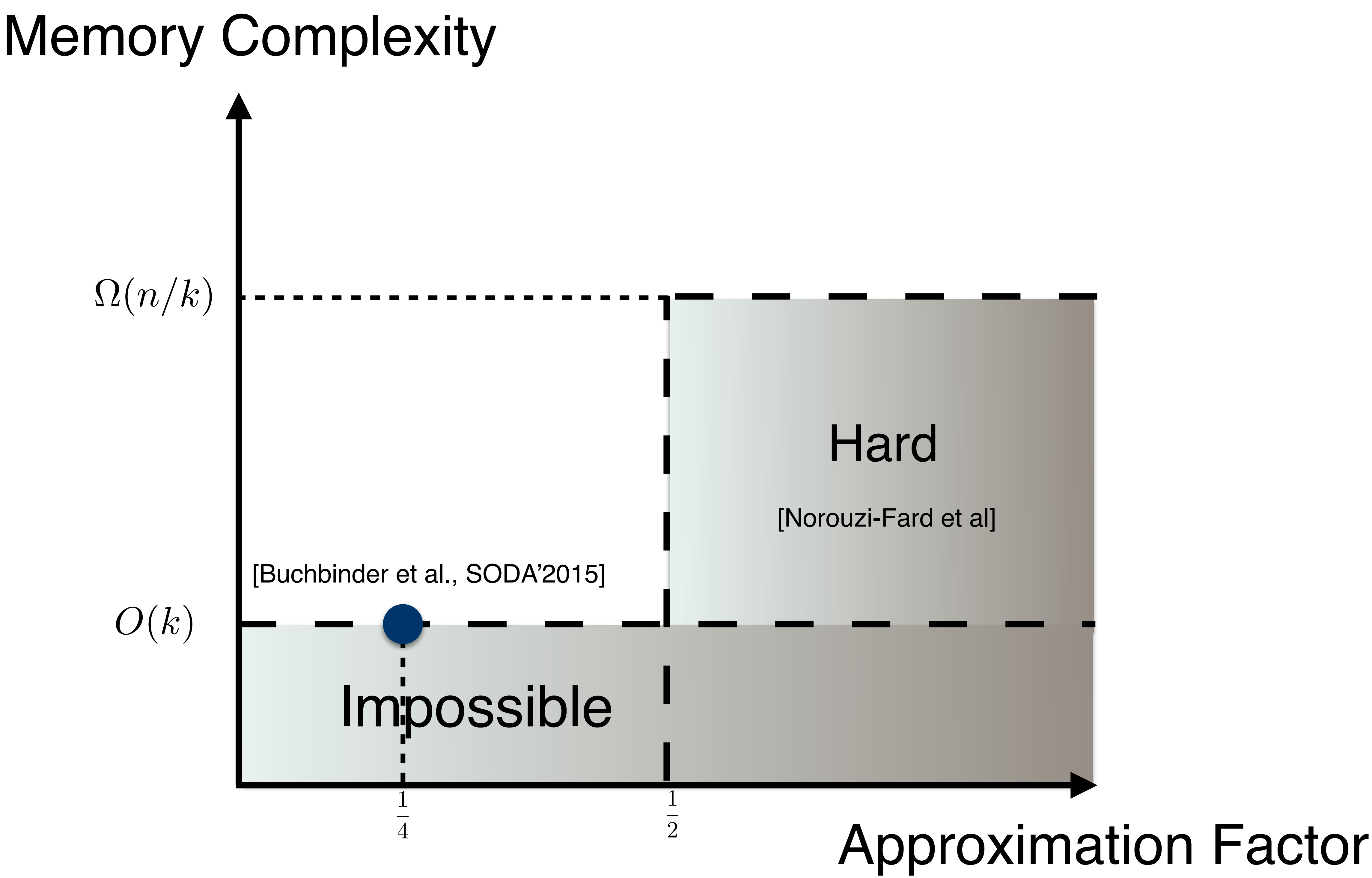
"Submodular Streaming in All Its Glory: Tight Approximation, Minimum Memory and Low Adaptive Complexity", ICML'19

Yale

Streaming Algorithms: Cardinality Constraint

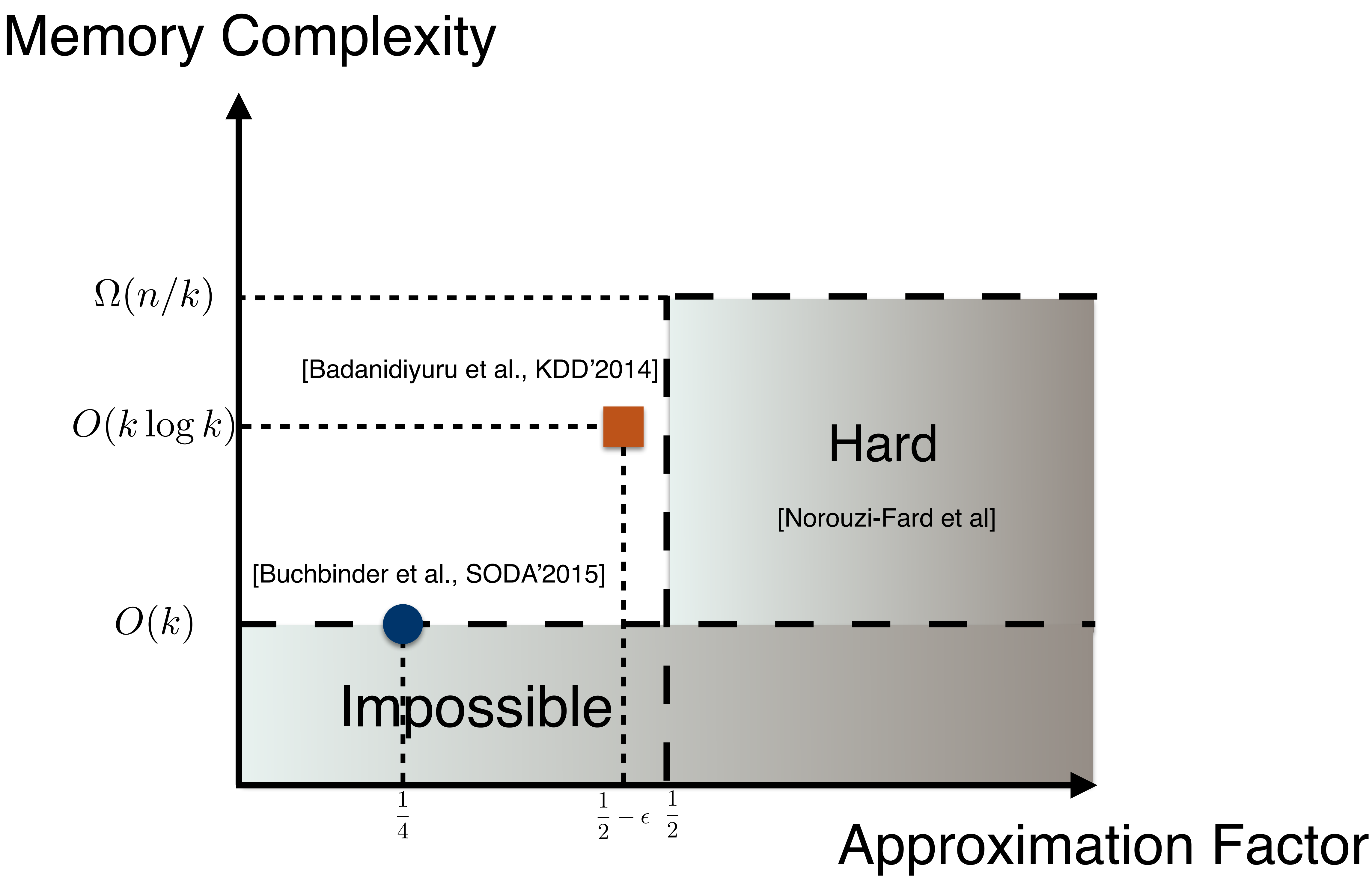


Streaming Algorithms: Cardinality Constraint



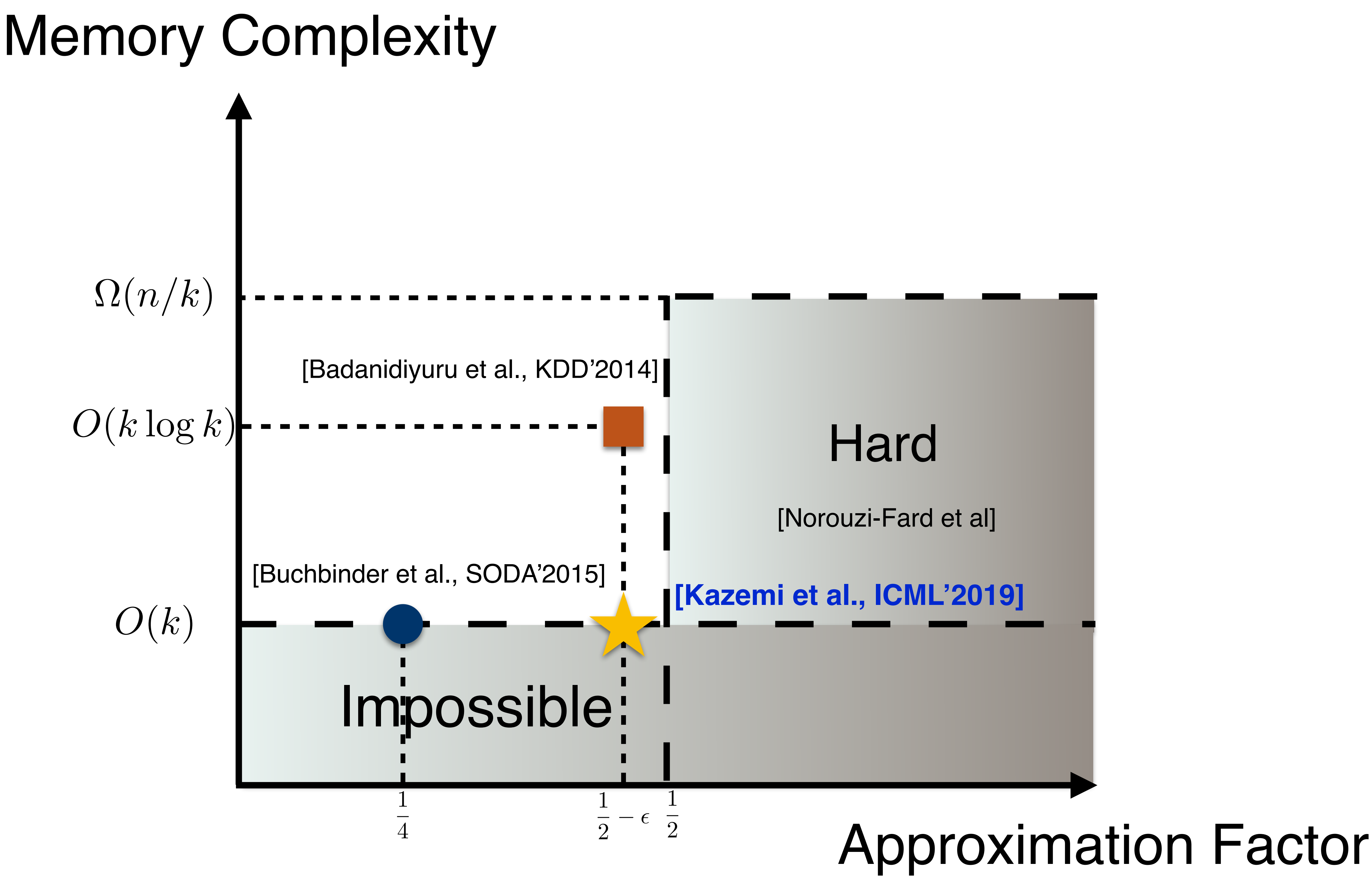
Yale

Streaming Algorithms: Cardinality Constraint

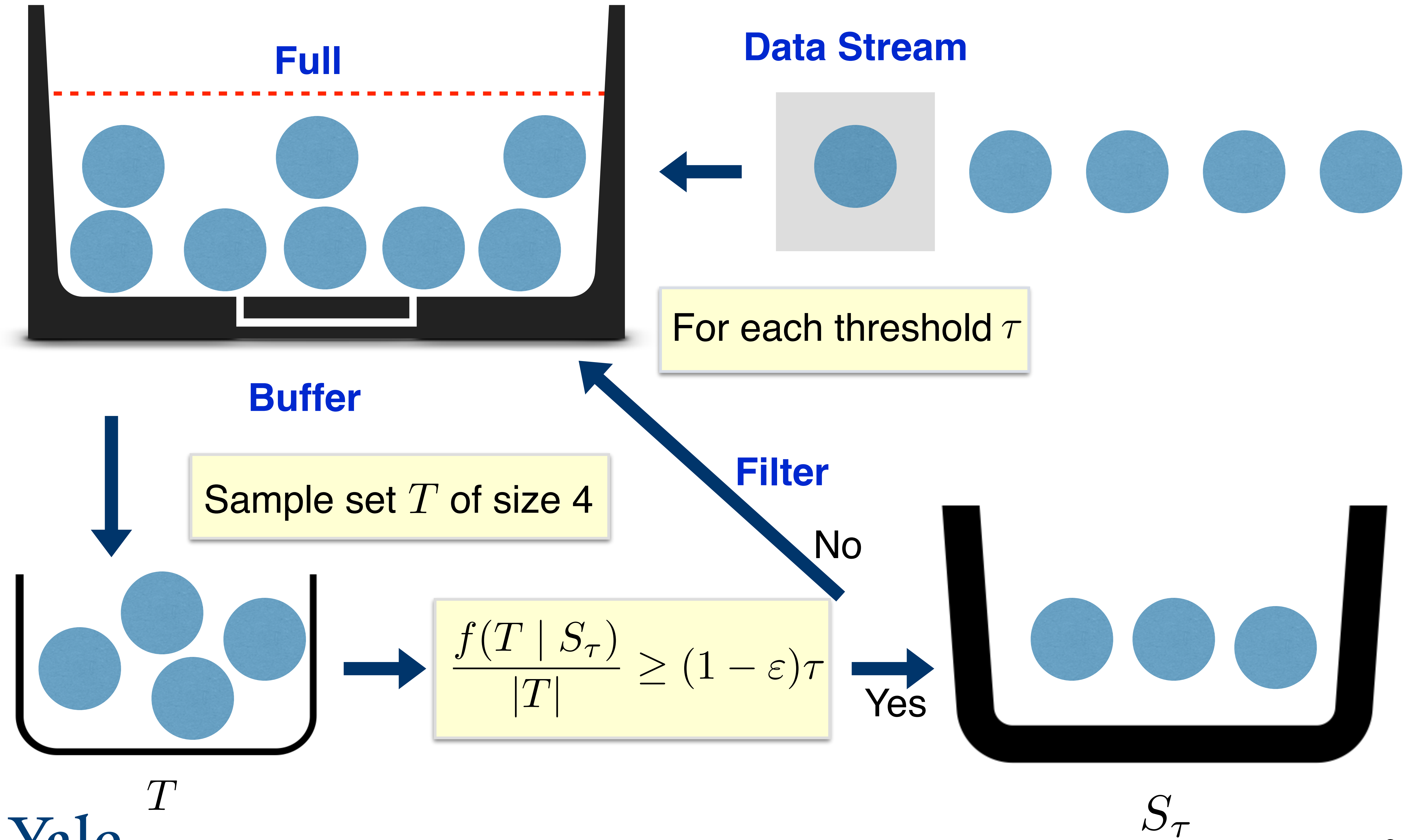


Yale

Streaming Algorithms: Cardinality Constraint

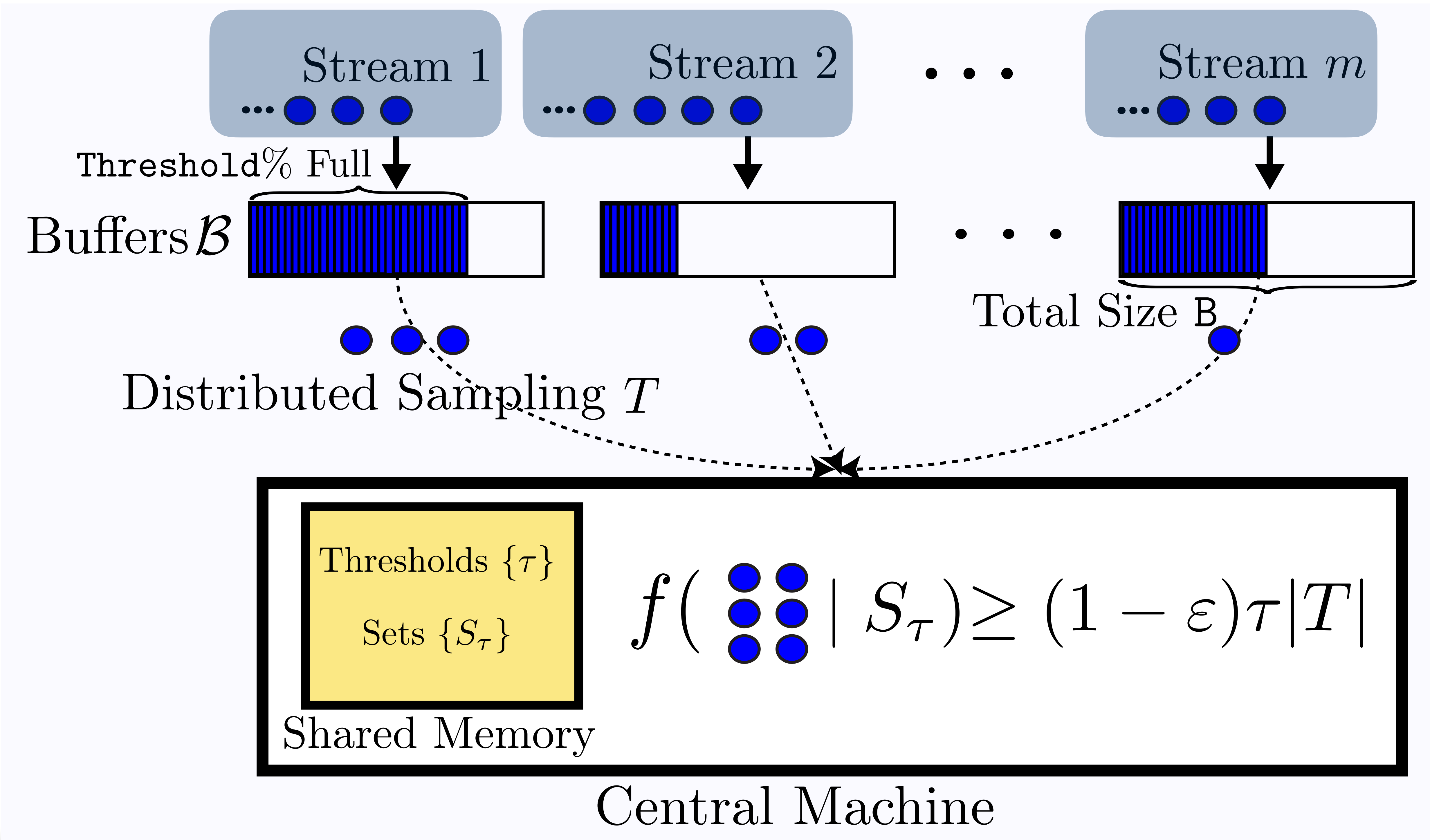


BATCH-SIEVE-STREAMING++



Yale

Multi-source Setting



Thank

You !