Streaming *k*-submodular Maximization under Noise subject to Size Constraint

Lan N. Nguyen, My T. Thai

University of Florida

k-submodular maximization s.t. size constraint

- k-submodular function is a generalization of submodular function
  - Submodular set function: input is a single subset V
    $$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$$
  - k-submodular function: input is k disjoint subsets of V
    $$f(\mathbf{x}) + f(\mathbf{y}) \geq f(\mathbf{x} \sqcup \mathbf{y}) + f(\mathbf{x} \sqcap \mathbf{y})$$
    - $\mathbf{x} = (X_1, \dots, X_k)$ and $\mathbf{y} = (Y_1, \dots, Y_k)$
    - $\mathbf{x} \sqcup \mathbf{y} = (Z_1, \dots, Z_k)$ where $Z_i = X_i \cup Y_i \setminus (\bigcup_{j \neq i} X_j \cup Y_j)$
    - $\mathbf{x} \sqcap \mathbf{y} = (X_1 \cap Y_1, \dots, X_k \cap Y_k)$
- k-submodular maximization s.t. size constraint (**M*k*SC**)
  - V – a finite set of elements, B – a positive integer.
  - $(k+1)^V$ - a family of k disjoint subsets of V
  - $f: (k+1)^V \to \mathbb{R}^+$ - a k-submodular function.

Find $\mathbf{s} = (S_1, \dots, S_k)$ s.t. $|\mathbf{s}| = |\bigcup_{i \leq k} S_i| \leq B$ that maximizes $f(\mathbf{s})$

k-submodular maximization s.t. size constraint

- Applications:
  - Influence maximization with k topics/products
  - Sensor placement with k kinds of sensors
  - Coupled Feature Selection.

- Existing solutions (*)
  - Greedy: 2 approximation ratio, $O(knB)$ query complexity
  - "Lazy" Greedy: 2 approximation ratio, $O(k(n-B)\log B \log\frac{B}{\delta})$ query complexity with probability at least $1-\delta$

(*) Ohsaka, Naoto, and Yuichi Yoshida. "Monotone k-submodular function maximization with size constraints." *Advances in Neural Information Processing Systems*. 2015.

Practical Challenges

- Noisy evaluation.
  - In many applications (e.g. Influence Maximization), obtaining exact value for $f(\mathbf{s})$ is impractical.
  - f can only be queried through a noisy version F

$$(1-\epsilon)f(\mathbf{s}) \leq F(\mathbf{s}) \leq (1+\epsilon)f(\mathbf{s}) \text{ for all } \mathbf{s} \in (k+1)^V$$

- Streaming.
  - Algorithms are required to take only one single pass over V
    - Produce solutions in a timely manner.
    - Avoid excessive storage in memory.

Our contribution

- Two streaming algorithms for **MkSC – DStream** & **RStream**
  - Take only 1 single scan over V
  - Access F instead of f
  - Performance guarantee:
    - Approximation ratio $f(\mathbf{s})/f(\mathbf{o})$: $\mathbf{o}$ - optimal solution.
    - Query and memory complexity

- Experimental Evaluation
  - Influence maximization with k topics.
  - Sensor placement with k kinds of sensor.

DStream

- Obtain o such that $f(o) \geq o \times B \geq f(o)/(1+\gamma)$
  - Using lazy estimation (*)

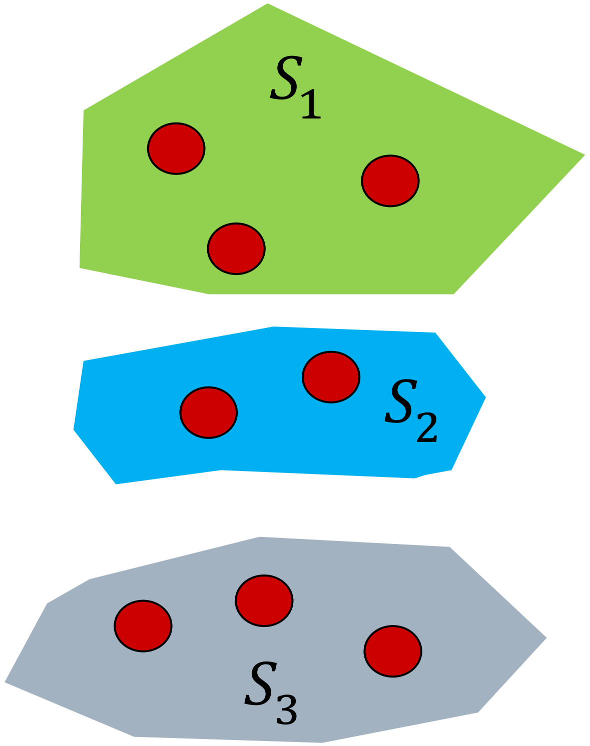
- For a new element e, if $|\mathbf{s}| < B$

Find

e

$$\max_{i \leq k} F(\mathbf{s} \sqcup (e, i))$$

Disjoint subsets obtained by putting e into S_i



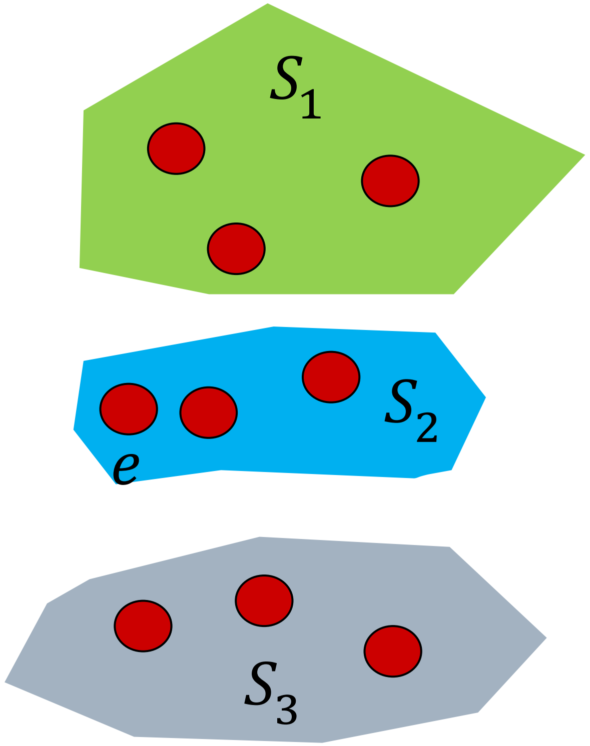
(*) Badanidiyuru, Ashwinkumar, et al. "Streaming submodular maximization: Massive data summarization on the fly." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014.

DStream

- Obtain o such that $f(o) \geq o \times B \geq f(o)/(1+\gamma)$
  - Using lazy estimation (*)

- For a new element e, if $|\mathbf{s}| < B$

Putting e to S_i if $\frac{F(\mathbf{s} \sqcup (e,i))}{1-\epsilon} \geq (|\mathbf{s}| + 1)\frac{o}{M}$



(*) Badanidiyuru, Ashwinkumar, et al. "Streaming submodular maximization: Massive data summarization on the fly." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014.

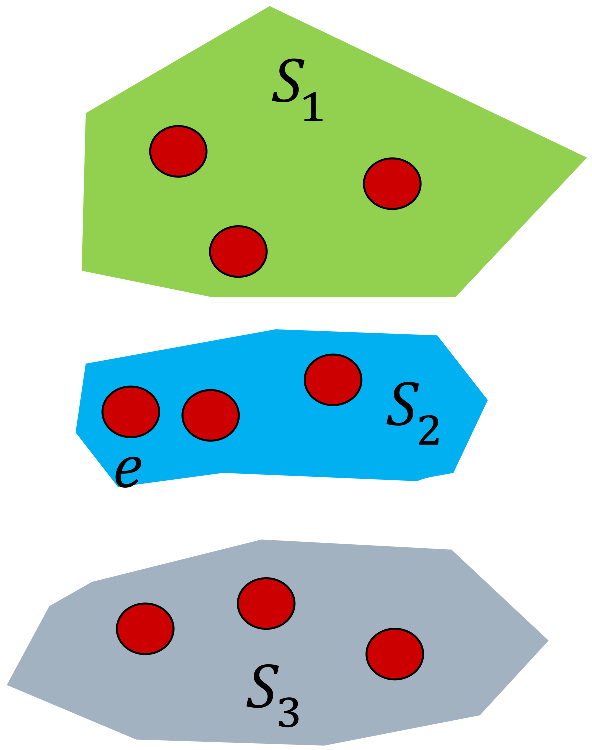
DStream

- Obtain o such that $f(o) \geq o \times B \geq f(o)/(1+\gamma)$
  - Using lazy estimation (*)

- For a new element e, if $|\mathbf{s}| < B$

Putting e to S_i if $\frac{F(\mathbf{s} \sqcup (e,i))}{1-\epsilon} \geq (|\mathbf{s}|+1)\frac{o}{M}$

Largest possible value of $f(\mathbf{s} \sqcup (e,i))$



(*) Badanidiyuru, Ashwinkumar, et al. "Streaming submodular maximization: Massive data summarization on the fly." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014.

DStream's performance guarantee

- $\mathbf{x} = (X_1, \dots, X_k)$ can also be understood as a vector $\mathbf{x}: V \to [k]$

	e_1	e_2	e_3	…	…	e_j	…	…	…	…
x	1	0	4	…	…	i	…	…	…	…

$$\mathbf{x}(e) = \begin{cases} i & \text{if } e \in X_i \\ 0 & \text{if } e \notin \bigcup_i X_i \end{cases}$$

DStream's performance guarantee

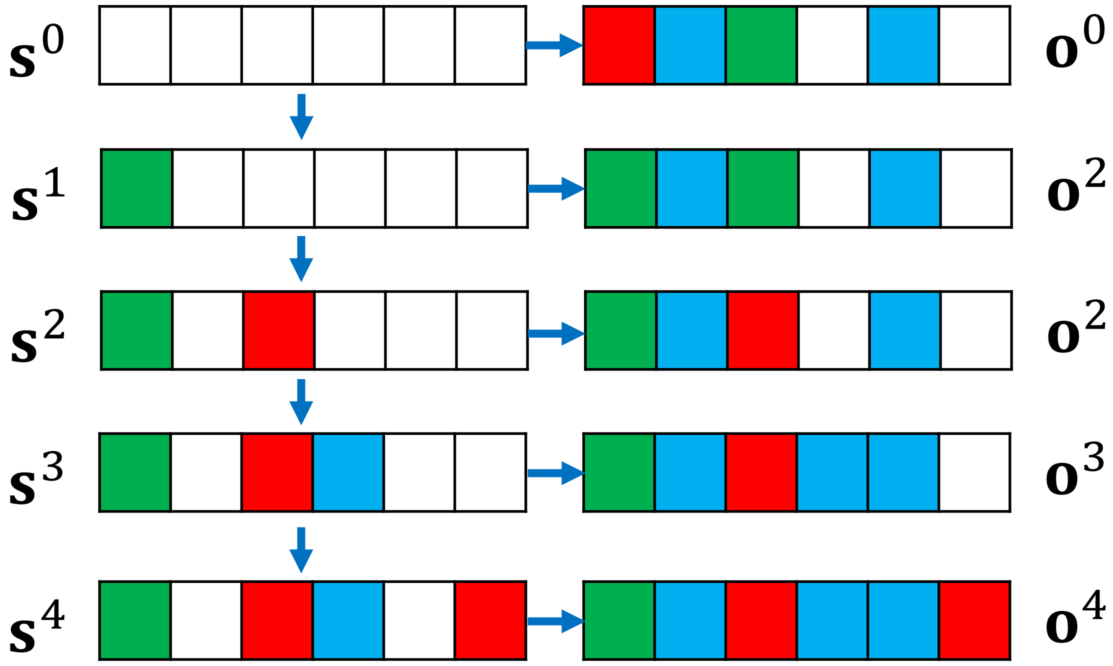
- $\mathbf{s}^0, \mathbf{s}^1, \ldots, \mathbf{s}^t$ - sequence of obtained solutions
  - $\mathbf{s}^i$ - obtained solution after adding i elements ($|\mathbf{s}^i| = i$)

- Construct a sequence $\mathbf{o}^0, \mathbf{o}^1, \ldots, \mathbf{o}^t$

$$\mathbf{o}^i = (\mathbf{o} \sqcup \mathbf{s}^i) \sqcup \mathbf{s}^i$$

$\mathbf{s}^i$	1	0	2	3	0	0	0	0
$\mathbf{o}$	2	1	2	0	0	3	0	1
$\mathbf{o}^i$	1	1	2	3	0	3	0	1

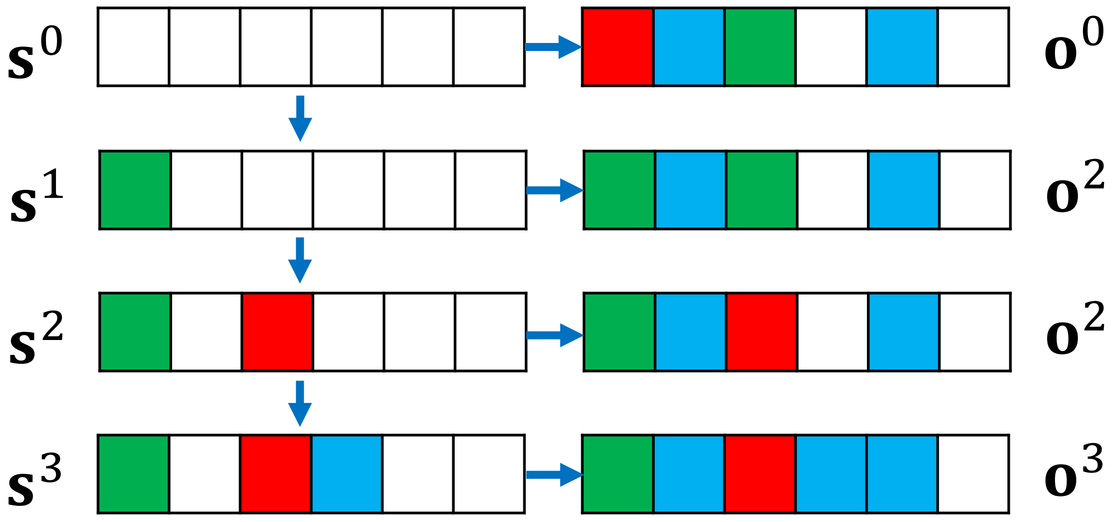
DStream's performance guarantee



- If in the end $|\mathbf{s}| = B$

$$f(\mathbf{s}) \geq \frac{1-\epsilon}{1+\epsilon} \frac{f(\mathbf{o})}{(1+\gamma)M}$$

DStream's performance guarantee



- If in the end $|\mathbf{s}| = t < B$, with f is **monotone**.
  - Establish recursive relationship between o^j, s^j

$$f(\mathbf{o}^{j-1}) + f(\mathbf{s}^{j-1}) \le f(\mathbf{o}^j) + \frac{1+\epsilon}{1-\epsilon} f(\mathbf{s}^j)$$

  - Bound $f(\mathbf{o}) - f(\mathbf{o}^t)$ $(*)$

$$f(\mathbf{o}) - f(\mathbf{o}^t) \le \frac{1+\epsilon+2B\epsilon}{1-\epsilon} f(\mathbf{s})$$

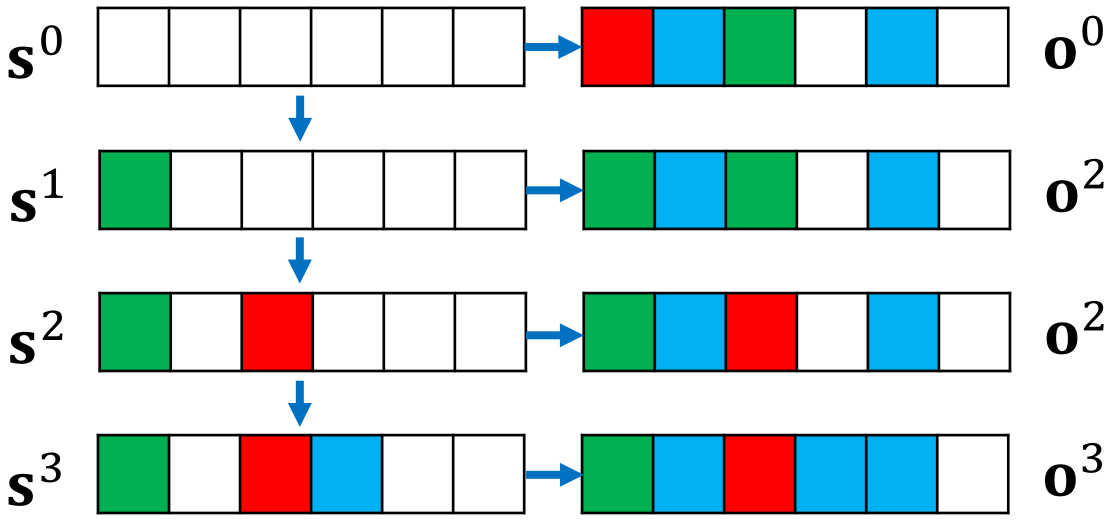
  - Bound $f(\mathbf{o}^t) - f(\mathbf{s})$ $(**)$

$$f(\mathbf{o}^t) - f(\mathbf{s}) \le \frac{1}{M} f(\mathbf{o}) + \frac{2B\epsilon}{1-\epsilon} f(\mathbf{s})$$

  - Discard $f(\mathbf{o}^t)$ by combining $(*)$ and $(**)$

$$f(\mathbf{o}) \le \frac{M}{M-1} \frac{2+4B\epsilon}{1-\epsilon} f(\mathbf{s})$$

DStream's performance guarantee



- If in the end $|\mathbf{s}| = t < B$, with f is **non-monotone**.
  - f is pairwise monotone

$$\Delta_{e,i} f(\mathbf{x}) + \Delta_{e,j} f(\mathbf{x}) \geq 0$$

  - Using the same framework as the **monotone** case but with different "math"

$$f(\mathbf{o}) \geq \frac{M}{M-1} \frac{(1+\epsilon)(3+3\epsilon+6B\epsilon)}{(1-\epsilon)^2} f(\mathbf{s})$$

DStream

Algorithm 2 DSTREAM

Input $V, F, k, B, M > 1, \gamma > 0$

$\Delta_u = \Delta_l = \Delta = 0; t_j = 0 \ \forall j \in \mathbb{Z}^+$
for each e in V **do**
  $\Delta = \max\left(\Delta, \max_{j \in [k]} F(\langle e, j \rangle)\right)$
  $\Delta_u = \Delta/(1-\epsilon); \Delta_l = \Delta/\left((1+\epsilon)(1+\gamma)\right)$
  $O = \{(1+\gamma)^j \mid \frac{\Delta_l}{B \cdot M} \le (1+\gamma)^j \le (1+\epsilon)\Delta_u\}$
  for each j that $(1+\gamma)^j \in O$ **do**
    $o = M(1+\gamma)^j$
    if $t_j < B$ **then**
      $i = argmax_{j' \in [k]} F(\mathbf{s}_j^{t_j} \sqcup \langle e, j' \rangle)$
      if $\frac{F(\mathbf{s}_j^{t_j} \sqcup \langle e, i \rangle)}{1-\epsilon} \ge (t_j + 1)\frac{o}{M}$ **then**
        $\mathbf{s}_j^{t_j+1} = \mathbf{s}_j^{t_j} \sqcup \langle e, i \rangle$
        $t_j = t_j + 1$

Return $argmax_{\mathbf{s}_j^{t_j}; j \in O} F(\mathbf{s}_j^{t_j})$ if f is **monotone**; $argmax_{\mathbf{s}_j^i; i \le t_j, j \in O} F(\mathbf{s}_j^i)$ if f is **non-monotone**.

Lazy estimation to obtain o

- $f(\mathbf{o}) \in [\Delta_l, B \times \Delta_u]$
- o can be obtained by a value of $(1+\gamma)^j \in [\frac{\Delta_l}{B}, M(1+\epsilon)\Delta_u]$

Query complexity

$$O\left(\frac{nk}{\gamma}\log\left(\frac{(1+\epsilon)(1+\gamma)}{1-\epsilon}BM\right)\right)$$

Memory complexity

$$O\left(\frac{B}{\gamma}\log\left(\frac{(1+\epsilon)(1+\gamma)}{1-\epsilon}BM\right)\right)$$

DStream

Algorithm 2 DSTREAM

Input $V, F, k, B, M > 1, \gamma > 0$

$\Delta_u = \Delta_l = \Delta = 0;\ t_j = 0\ \forall j \in \mathbb{Z}^+$
for each e in V **do**
  $\Delta = \max\left(\Delta, \max_{j\in[k]} F(\langle e, j\rangle)\right)$
  $\Delta_u = \Delta/(1-\epsilon);\ \Delta_l = \Delta/\left((1+\epsilon)(1+\gamma)\right)$
  $O = \{(1+\gamma)^j \mid \frac{\Delta_l}{B\cdot M} \le (1+\gamma)^j \le (1+\epsilon)\Delta_u\}$
  for each j that $(1+\gamma)^j \in O$ **do**
    $o = M(1+\gamma)^j$
    if $t_j < B$ **then**
      $i = argmax_{j'\in[k]} F(\mathbf{s}_j^{t_j} \sqcup \langle e, j'\rangle)$
      if $\frac{F(\mathbf{s}_j^{t_j} \sqcup \langle e,i\rangle)}{1-\epsilon} \ge (t_j+1)\frac{o}{M}$ **then**
        $\mathbf{s}_j^{t_j+1} = \mathbf{s}_j^{t_j} \sqcup \langle e, i\rangle$
        $t_j = t_j + 1$

Return $argmax_{\mathbf{s}_j^{t_j};j\in O} F(\mathbf{s}_j^{t_j})$ if f is **monotone**;
$argmax_{\mathbf{s}_j^i;i\le t_j,j\in O} F(\mathbf{s}_j^i)$ if f is **non-monotone**.

Approximation ratio

$$\frac{1+\epsilon}{1-\epsilon} \min_{x\in(1,M]} \max(a(x), b(x))$$

If f is **monotone**

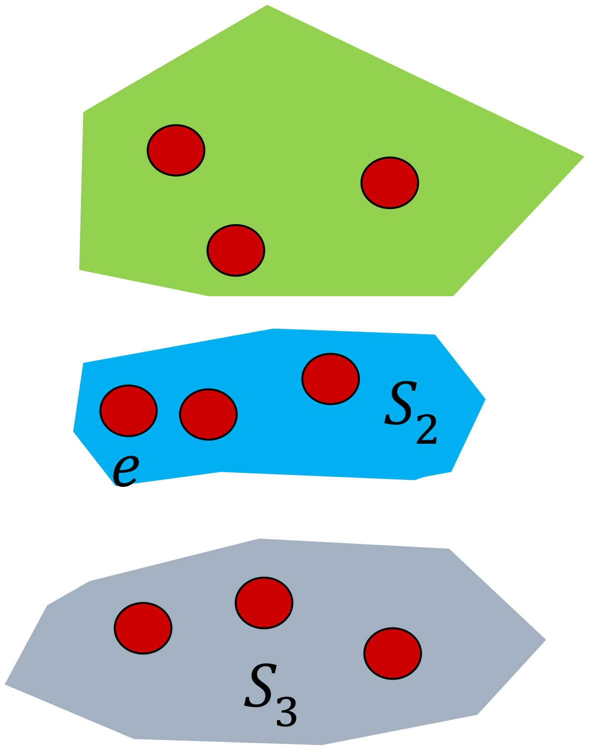
- $a(x) = \frac{(1+\gamma)(1+\epsilon)}{1-\epsilon} x$
- $b(x) = \frac{2+4B\epsilon}{1-\epsilon} \frac{x}{x-1}$

If f is **non-monotone**

- $a(x) = \frac{(1+\gamma)(1+\epsilon)}{1-\epsilon} x$
- $b(x) = \frac{(1+\epsilon)(3+3\epsilon+6B\epsilon)}{(1-\epsilon)^2} \frac{x}{x-1}$

DStream's weakness

Putting e to S_i if $\frac{F(\mathbf{s} \sqcup (e,i))}{1-\epsilon} \geq (|\mathbf{s}| + 1)\frac{o}{M}$



What if $f(\mathbf{s}) \geq (|\mathbf{s}| + 1)\frac{o}{M}$?

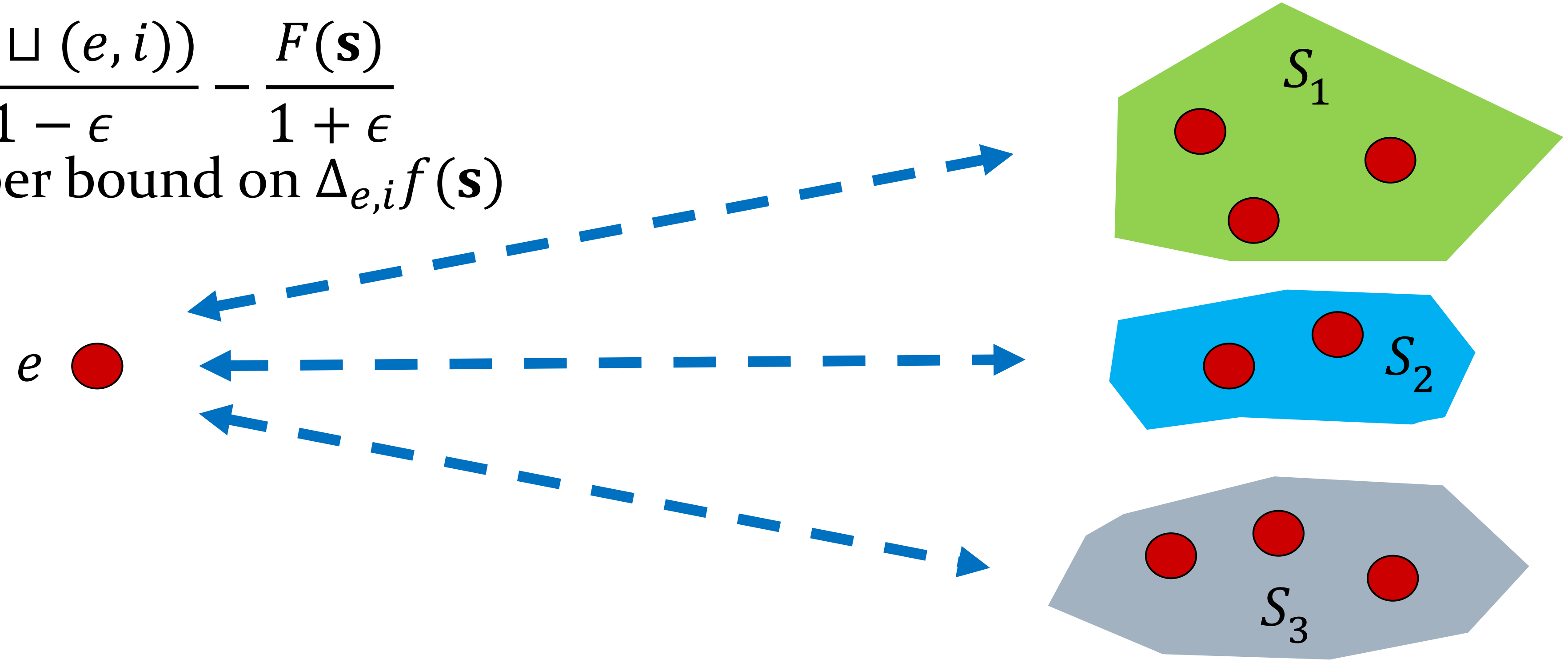
- e may have no contribution to $\mathbf{s}$
- Better consider marginal gain

RStream

➢ For a new element e, if $|\mathbf{s}| < B$

$$d_i = \frac{F(\mathbf{s} \sqcup (e,i))}{1-\epsilon} - \frac{F(\mathbf{s})}{1+\epsilon}$$

- d_i is an upper bound on $\Delta_{e,i} f(\mathbf{s})$

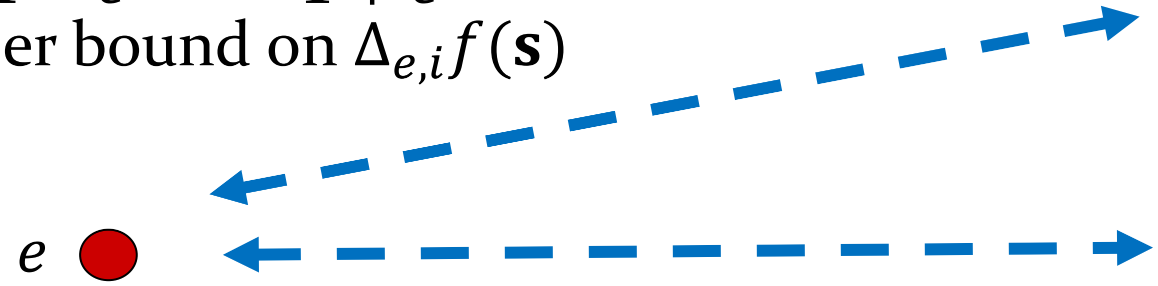


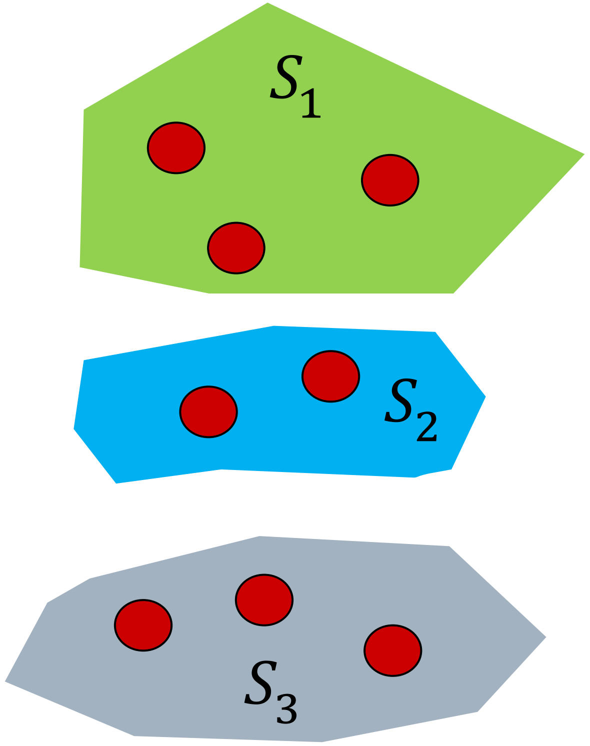
RStream

- For a new element e, if $|\mathbf{s}| < B$

$$d_i = \frac{F(\mathbf{s} \sqcup (e,i))}{1-\epsilon} - \frac{F(\mathbf{s})}{1+\epsilon}$$

- d_i is an upper bound on $\Delta_{e,i} f(\mathbf{s})$





- Filter out S_i that $d_i \le \frac{o}{M}$
  - $d_i = 0$ if $d_i \le \frac{o}{M}$
  - Otherwise d_i keeps its value
- Randomly put e into S_i with probability

$$d_i^{T-1} / \sum_j d_j^{T-1}$$

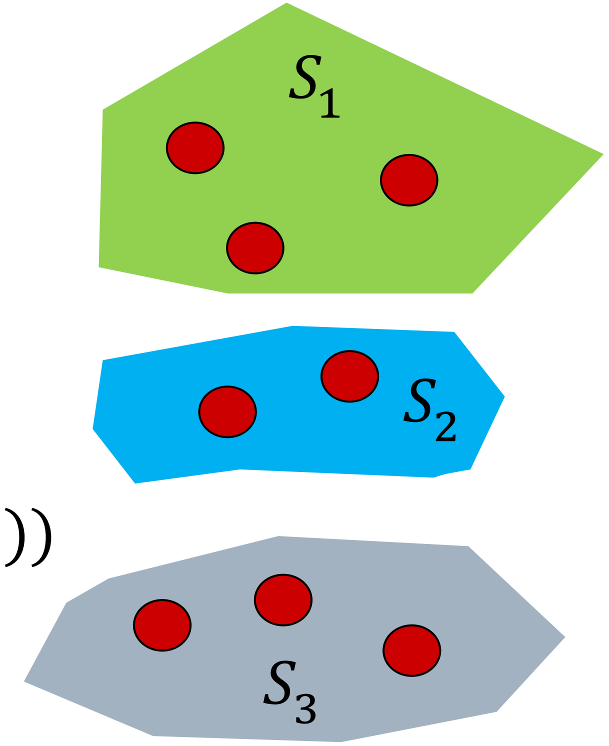
- $T = \left|\left\{ j : d_j \ge \frac{o}{M} \right\}\right|$

RStream

$$d_i = \frac{F(\mathbf{s} \sqcup (e,i))}{1-\epsilon} - \frac{F(\mathbf{s})}{1+\epsilon}$$

- d_i is an upper bound on $\Delta_{e,i} f(\mathbf{s})$

e



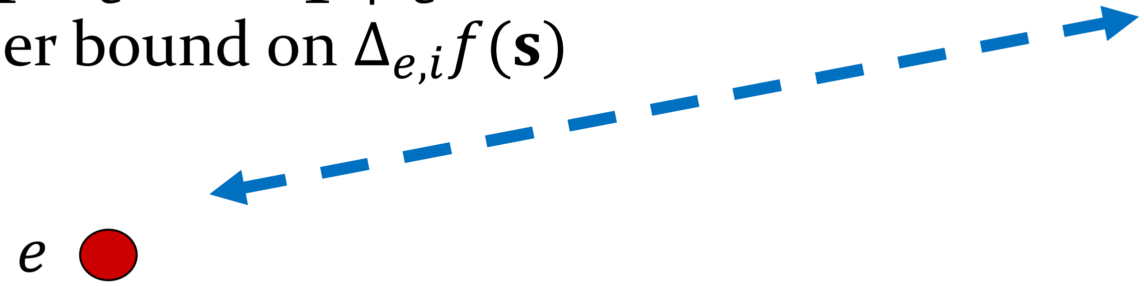
What if $F(\mathbf{s}) \approx f(\mathbf{s}) = f(\mathbf{s} \sqcup (e,i)) \approx F(\mathbf{s} \sqcup (e,i))$

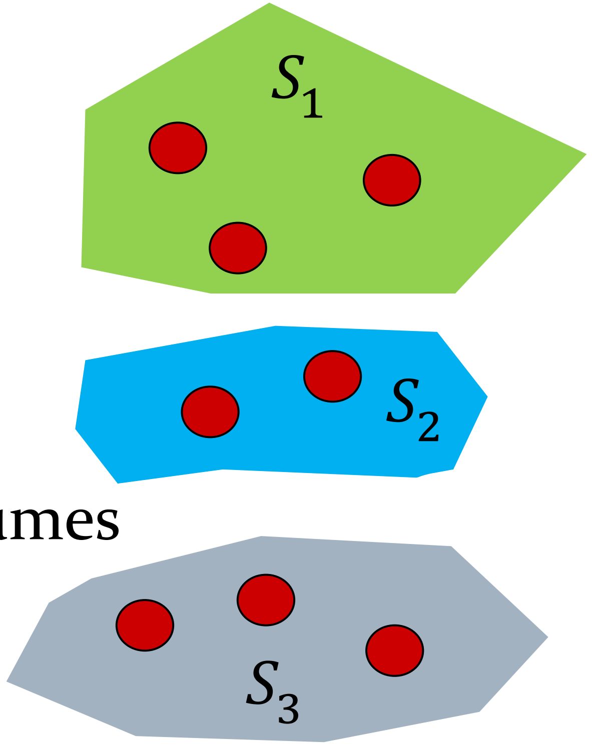
- e has no contribution
- But $d_i \approx \frac{2\epsilon}{1-\epsilon^2} f(\mathbf{s}) \geq \frac{o}{M}$

RStream

$$d_i = \frac{F(\mathbf{s} \sqcup (e,i))}{1-\epsilon} - \frac{F(\mathbf{s})}{1+\epsilon}$$

- d_i is an upper bound on $\Delta_{e,i} f(\mathbf{s})$





(Denoise) Run multiple instances, each instance assumes F is less noisy than it is.

$$\boldsymbol{d_{i,\epsilon'}} = \frac{F(\mathbf{s} \sqcup (e,i))}{1-\boldsymbol{\epsilon'}} - \frac{F(\mathbf{s})}{1+\boldsymbol{\epsilon'}}$$

where $\epsilon' = 0, \frac{\epsilon}{\eta-1}, \frac{2\epsilon}{\eta-1}, \ldots, \epsilon$

η – adjustable parameter, controlling number of instances

Algorithm 3 RSTREAM

Input $F, k, B, M > 1, \gamma > 0, \eta \geq 1$

$\Delta_u = \Delta_l = \Delta = 0; t_{j,\epsilon'} = 0 \; \forall j \in \mathbb{Z}^+, \epsilon' \in \mathbb{R}^+$

for each e in V **do**

  $\Delta = \max\left(\Delta, \max_{j \in [k]} F(\langle e, j \rangle)\right)$

  $\Delta_u = \frac{(1+\epsilon)^2 + 4\epsilon B}{(1-\epsilon^2)(1-\epsilon)} \Delta; \Delta_l = \Delta/((1+\epsilon)(1+\gamma))$

  $O = \{(1+\gamma)^j \mid \frac{\Delta_l}{B \cdot M} \leq (1+\gamma)^j \leq \Delta_u\}$

  for each j that $(1+\gamma)^j \in O$ **do**

    $o = M(1+\gamma)^j$

    for each $\epsilon' = \epsilon, \frac{(\eta-2)\epsilon}{\eta-1}, \frac{(\eta-3)\epsilon}{\eta-1}, \ldots 0$ **do**

      if $t_{j,\epsilon'} < B$ **then**

        for each $i \in [k]$ **do**

          $d_i = \frac{F(\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}} \sqcup \langle e, i \rangle)}{1-\epsilon'} - \frac{F(\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}})}{1+\epsilon'}$

          $d_i = 0$ if $d_i < \frac{o}{M}$, d_i otherwise

        $T =$ no. d_i that $d_i > 0$

        $D = \sum_{i \in [k]} d_i^{T-1}$

        if $t_{j,\epsilon'} < B$ and $T > 0$ **then**

          if $T = 1$ **then**

            $i =$ the only one that $d_i > 0$

          else

            $i =$ selected with prob. d_i^{T-1}/D

          $\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}+1} = \mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}} \sqcup \langle e, i \rangle$

          $t_{j,\epsilon'} = t_{j,\epsilon'} + 1$

Return $argmax_{\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}}, j \in O} F(\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}})$ if f is **monotone**;

$argmax_{\mathbf{s}_{j,\epsilon'}^{i} | i < t_{j,\epsilon'}, j \in O} F(\mathbf{s}_{j,\epsilon'}^{i})$ if f is **non-monotone**;

(**Denoise**) Run multiple instances, each instance assumes F is less noisy than it actually is.

Algorithm 3 RSTREAM

Input $F, k, B, M > 1, \gamma > 0, \eta \geq 1$

```
Δ_u = Δ_l = Δ = 0; t_{j,ε'} = 0 ∀j ∈ Z^+, ε' ∈ R^+
for each e in V do
    Δ = max(Δ, max_{j∈[k]} F(⟨e, j⟩))
    Δ_u = ((1+ε)^2 + 4εB) / ((1−ε^2)(1−ε)) Δ; Δ_l = Δ/((1+ε)(1+γ))
    O = {(1+γ)^j | Δ_l/(B·M) ≤ (1+γ)^j ≤ Δ_u}
    for each j that (1+γ)^j ∈ O do
        o = M(1+γ)^j
        for each ε' = ε, (η−2)ε/(η−1), (η−3)ε/(η−1), ...0 do
            if t_{j,ε'} < B then
                for each i ∈ [k] do
                    d_i = F(s_{j,ε'}^{t_{j,ε'}} ⊔ ⟨e,i⟩)/(1−ε') − F(s_{j,ε'}^{t_{j,ε'}})/(1+ε')
                    d_i = 0 if d_i < o/M, d_i otherwise
                T = no. d_i that d_i > 0
                D = Σ_{i∈[k]} d_i^{T−1}
                if t_{j,ε'} < B and T > 0 then
                    if T = 1 then
                        i = the only one that d_i > 0
                    else
                        i = selected with prob. d_i^{T−1}/D
                    s_{j,ε'}^{t_{j,ε'}+1} = s_{j,ε'}^{t_{j,ε'}} ⊔ ⟨e, i⟩
                    t_{j,ε'} = t_{j,ε'} + 1
```

Return $argmax_{\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}}, j\in O} F(\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}})$ if f is **monotone**;
$argmax_{\mathbf{s}_{j,\epsilon'}^{i} | i < t_{j,\epsilon'}, j\in O} F(\mathbf{s}_{j,\epsilon'}^{i})$ if f is **non-monotone**;

Lazy estimation: Δ_u is much larger than the one in DStream in order to bound d_is' value.

Query complexity

$$O(\frac{nk\eta}{\gamma}\log(\frac{((1+\epsilon)^2+4B\epsilon)(1+\gamma)}{(1-\epsilon)^2}BM))$$

Memory complexity

$$O(\frac{\eta B}{\gamma}\log(\frac{((1+\epsilon)^2+4B\epsilon)(1+\gamma)}{(1-\epsilon)^2}BM))$$

Algorithm 3 RSTREAM

Input $F, k, B, M > 1, \gamma > 0, \eta \geq 1$

$\Delta_u = \Delta_l = \Delta = 0; t_{j,\epsilon'} = 0\ \forall j \in \mathbb{Z}^+, \epsilon' \in \mathbb{R}^+$

for each e in V **do**

  $\Delta = \max\left(\Delta, \max_{j\in[k]} F(\langle e, j\rangle)\right)$

  $\Delta_u = \frac{(1+\epsilon)^2+4\epsilon B}{(1-\epsilon^2)(1-\epsilon)}\Delta; \Delta_l = \Delta/((1+\epsilon)(1+\gamma))$

  $O = \{(1+\gamma)^j \mid \frac{\Delta_l}{B\cdot M} \leq (1+\gamma)^j \leq \Delta_u\}$

  for each j that $(1+\gamma)^j \in O$ **do**

    $o = M(1+\gamma)^j$

    for each $\epsilon' = \epsilon, \frac{(\eta-2)\epsilon}{\eta-1}, \frac{(\eta-3)\epsilon}{\eta-1}, ...0$ **do**

      if $t_{j,\epsilon'} < B$ **then**

        for each $i \in [k]$ **do**

          $d_i = \frac{F(\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}} \sqcup \langle e,i\rangle)}{1-\epsilon'} - \frac{F(\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}})}{1+\epsilon'}$

          $d_i = 0$ if $d_i < \frac{o}{M}$, d_i otherwise

        $T =$ no. d_i that $d_i > 0$

        $D = \sum_{i\in[k]} d_i^{T-1}$

        if $t_{j,\epsilon'} < B$ and $T > 0$ **then**

          if $T = 1$ **then**

            $i =$ the only one that $d_i > 0$

          else

            $i =$ selected with prob. d_i^{T-1}/D

          $\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}+1} = \mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}} \sqcup \langle e, i\rangle$

          $t_{j,\epsilon'} = t_{j,\epsilon'} + 1$

Return $argmax_{\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}}, j\in O} F(\mathbf{s}_{j,\epsilon'}^{t_{j,\epsilon'}})$ if f is **monotone**;

$argmax_{\mathbf{s}^i_{j,\epsilon'} | i<t_{j,\epsilon'}, j\in O} F(\mathbf{s}^i_{j,\epsilon'})$ if f is **non-monotone**;

Approximation ratio

$$\frac{1+\epsilon}{1-\epsilon} \min_{x\in(1,M]} \max(a(x), b(x))$$

If f is **monotone**

- $a(x) = \frac{(1+\gamma)(1+\epsilon+2B\epsilon)}{1-\epsilon} x$
- $b(x) = \left(\frac{(1+\epsilon)^2+4B\epsilon}{1-\epsilon^2}\left(1-\frac{1}{k}\right)+1\right)\frac{kx}{kx-k-1}$

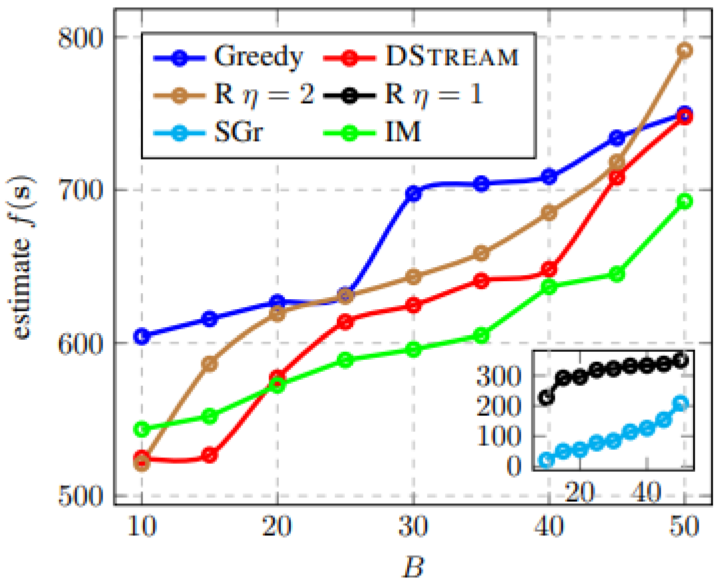
If f is **non-monotone**

- $a(x) = \frac{(1+\gamma)(1+\epsilon+2B\epsilon)}{1-\epsilon} x$
- $b(x) = \frac{(3k-2)(1+\epsilon)^2+(8k-8)B\epsilon}{(1-\epsilon)^2} \frac{x}{kx-k-2}$

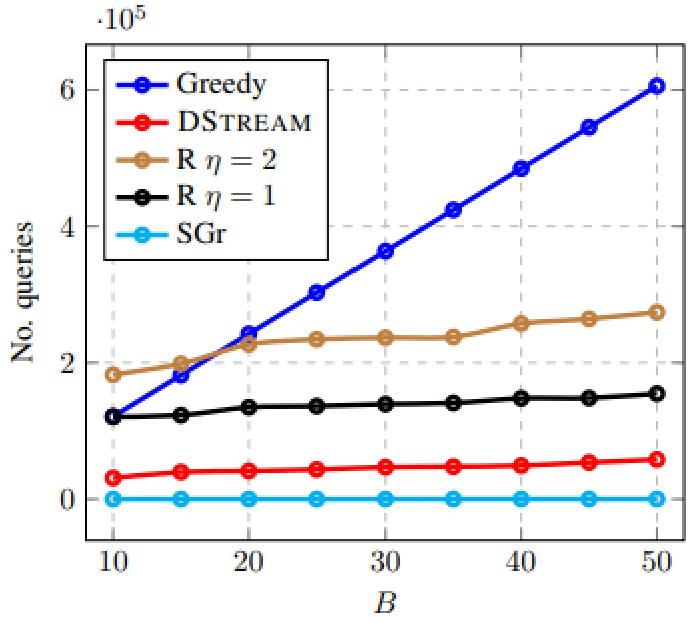
Experimental Evaluation

- Influence Maximization with k topics
  - k influence spread processes occur independently in a social network.
  - Find $S_1, \ldots, S_k$ that maximize the number of active users
    - An active user is a user who is activated by at least 1 topics.
    - S_i - a seed set of users who start spreading topic i
    - $|S_1 \cup \cdots \cup S_k| \leq B$
- Social network: Facebook dataset from SNAP
  - Leskovec, Jure, and Rok Sosič. "Snap: A general-purpose network analysis and graph-mining library." *ACM Transactions on Intelligent Systems and Technology (TIST)* 8.1 (2016): 1-20.
- Influence model: Linear Threshold
  - Kempe, David, Jon Kleinberg, and Éva Tardos. "Maximizing the spread of influence through a social network." *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003.

Influence Maximization with *k* topics



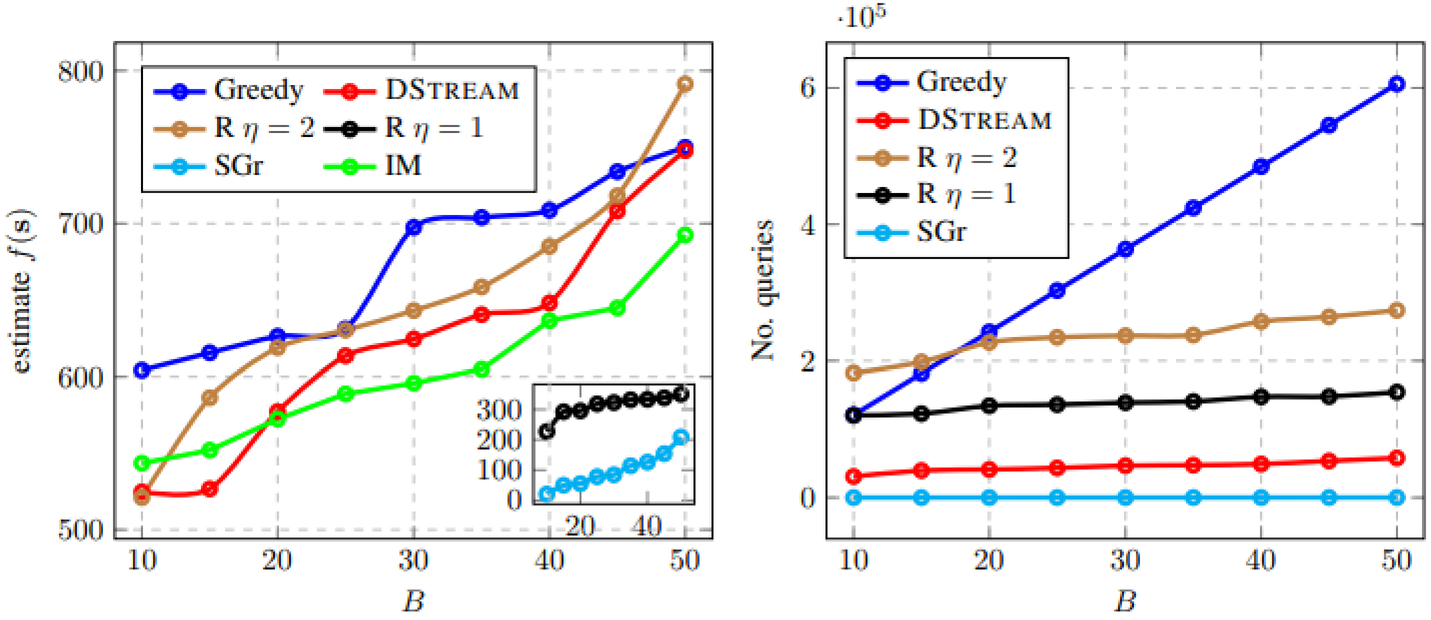
(a) Quality of solution



(b) No. queries

- Compared algorithms
  - **Greedy** (Ohsaka, Naoto, and Yuichi Yoshida et al. NIPS'15)
  - **IM**: randomly select 1 topic and solve classical Influence Maximization problem
  - **SGr:** simple streaming, pick e with prob. $\frac{B}{n}$ and put to S_i that maximizes $F(s \sqcup (e, i))$

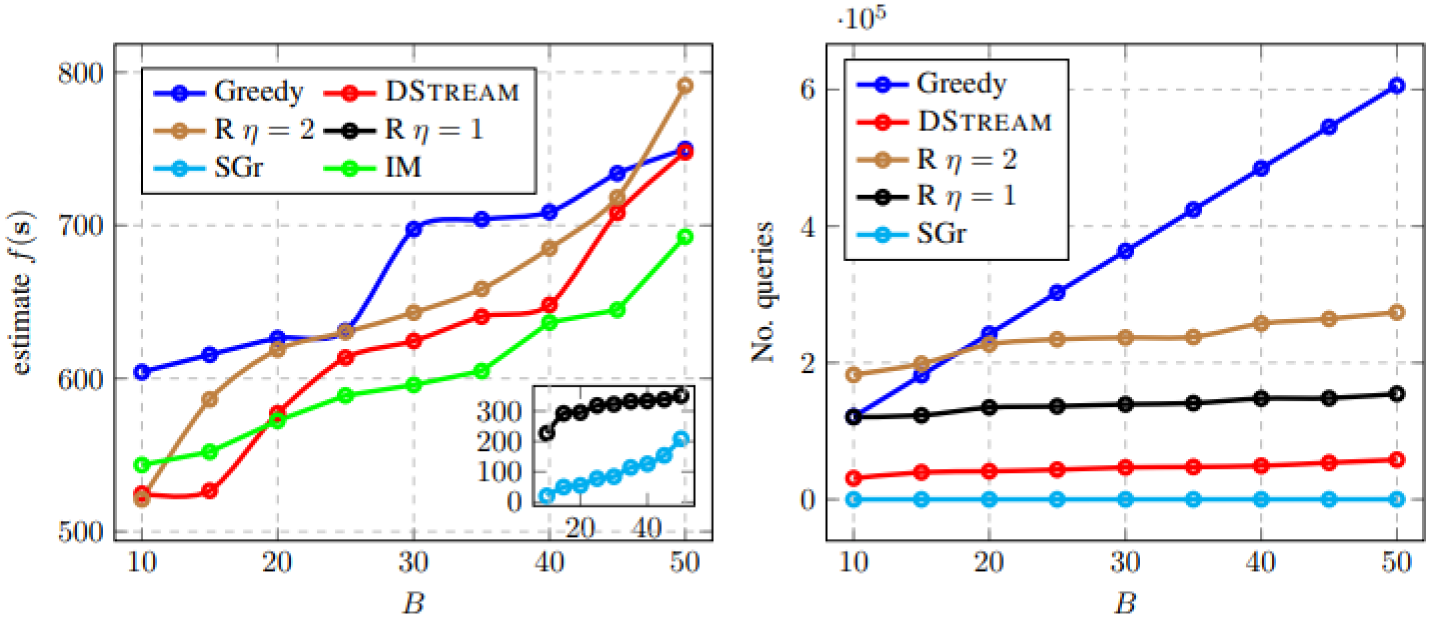
Influence Maximization with *k* topics



(a) Quality of solution (b) No. queries

- DStream and RStream ($\eta = 2$)
  - Returned solutions approximately to Greedy, outperformed IM in most cases.
  - Outperformed Greedy in # queries by a huge margin.

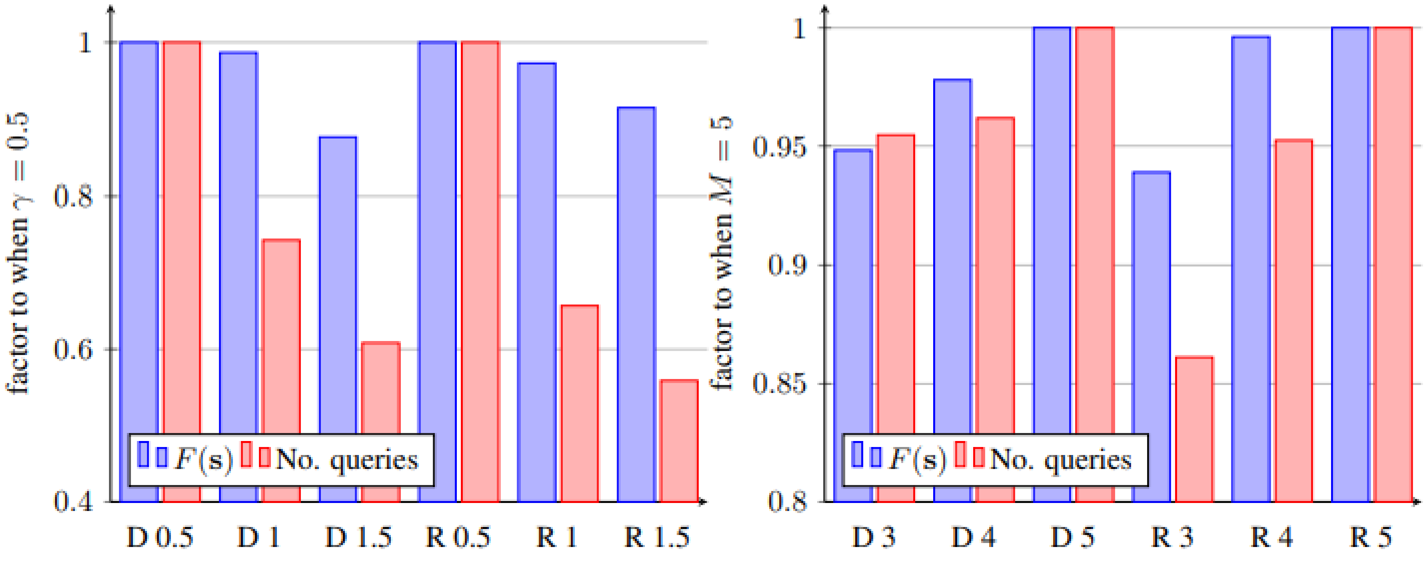
Influence Maximization with *k* topics



(a) Quality of solution (b) No. queries

- **Denoise** step helped RStream improve performance.
  - $\eta = 1$ causes RStream terminate prematurely and perform worse than DStream
  - $\eta = 2$ helps RStream improve solution quality but take 4 times more queries than DStream.

Influence Maximization with *k* topics



(a) Tradeoff w.r.t γ (b) Tradeoff w.r.t M

- The **larger** $\boldsymbol{\gamma}$ is, the **lower solution quality** and the **fewer queries** the algorithms obtained.
- The **smaller M** is, the **lower solution quality** and the **fewer queries** the algorithms obtained.

Conclusion

- We propose 2 streaming algorithms with theoretical performance guarantee to solve **MkSC** under noise.

- In comparison with Greedy, our algorithms
  - Take much fewer queries
  - Obtain comparable solutions in term of quality.

- Thanks! Questions?
  - lan.nguyen@ufl.edu