Fairness in Streaming Submodular Maximization over a Matroid Constraint

Marwa El Halabi — *Samsung - SAIT AI Lab, Montreal*
Federico Fusco — *Sapienza University of Rome*
Ashkan Norouzi-Fard — *Google Research*
Jakab Tardos — *Google*
Jakub Tarnawski — *Microsoft Research*

Fair Matroid Monotone Submodular Maximization

$$\max_{S \subseteq V}\{f(S) : S \in \mathcal{I}, \ell_c \le |S \cap V_c| \le u_c \text{ for all } c = 1, \cdots, C\} \quad \text{(FMMSM)}$$

Set function $f : 2^V \to \mathbb{R}_+$ is

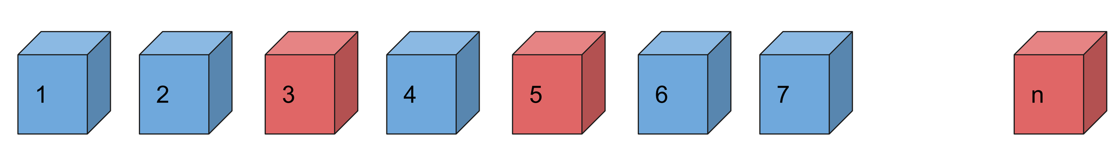
- **Monotone:** for any set X and element e,
$$f(X \cup \{e\}) \ge f(X)$$
- **Submodular:** for any sets $X \subseteq Y$ and element e
$$f(X \cup \{e\}) - f(X) \ge f(Y \cup \{e\}) - f(Y)$$

Matroid $\mathcal{I} \subseteq 2^V$ of rank k: non-empty family of sets satisfying

- Downward-closedness: if $A \subseteq B$ and $B \in \mathcal{I}$, then $A \in \mathcal{I}$
- Augmentation: if $A, B \in \mathcal{I}$ with $|A| < |B|$, then $A + e \in \mathcal{I}$ for some $e \in B$.

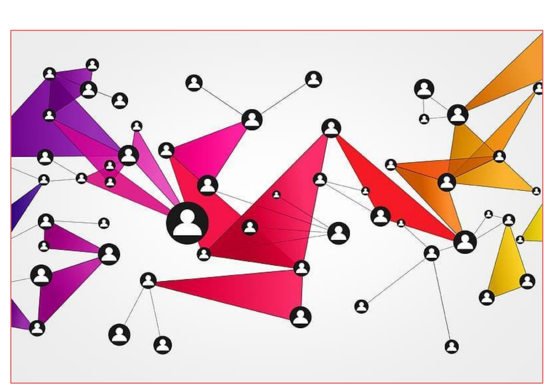
Examples: uniform matroid |S| ≤ k, partition matroid $|S \cap V_c| \le u_c$

Streaming setting: Elements arrive on a stream and we have limited memory.



Fairness: Solution should be balanced with respect to some sensitive attribute.

- Each element has a color c encoding the sensitive attribute.
- V is partitioned into C disjoint color groups $V = V_1 \cup \cdots \cup V_C$.
- We are given a lower ℓ_c and upper bound u_c (*not constants*) on the number of elements we can pick from each color c.



Applications: multiwinner voting, influence maximization, data summarization

Related work

Special case of FMMSM with cardinality constraint:

- Celis et al. [2018]: tight (1 - 1/e)-approximation in centralised setting.
- El Halabi et al. [2020]: one-pass streaming algorithms with
  - tight 1/2-approximation with exponential in k memory
  - 1/4-approximation with O(k) memory

Monotone submodular maximization over two matroid constraints:

- Garg et al., [2021]: 1/5.828-approximation one-pass streaming algorithm with O(k) memory.

Our Results

A tight 1/2 - approximation algorithm with exponential memory.

Theorem 1.1. *For any constant $\eta \in (0, 1/2)$, there exists a one-pass streaming $(1/2 - \eta)$-approximation algorithm for FMMSM that uses $2^{O(k^2 + k \log C)} \cdot \log \Delta$ memory, where $\Delta = \frac{\max_{e \in V} f(e)}{\min_{\{e \in V \mid f(e) > 0\}} f(e)}$.*

What if we want to Use Less Memory?

It is not possible to use efficient memory even if we make multiple passes.

Theorem 1.2. *Any (randomized) $o(\sqrt{\log C})$-pass streaming algorithm that determines the existence of a feasible solution for FMMSM with probability at least $2/3$ requires $\max(k, C)^{2-o(1)}$ memory.*

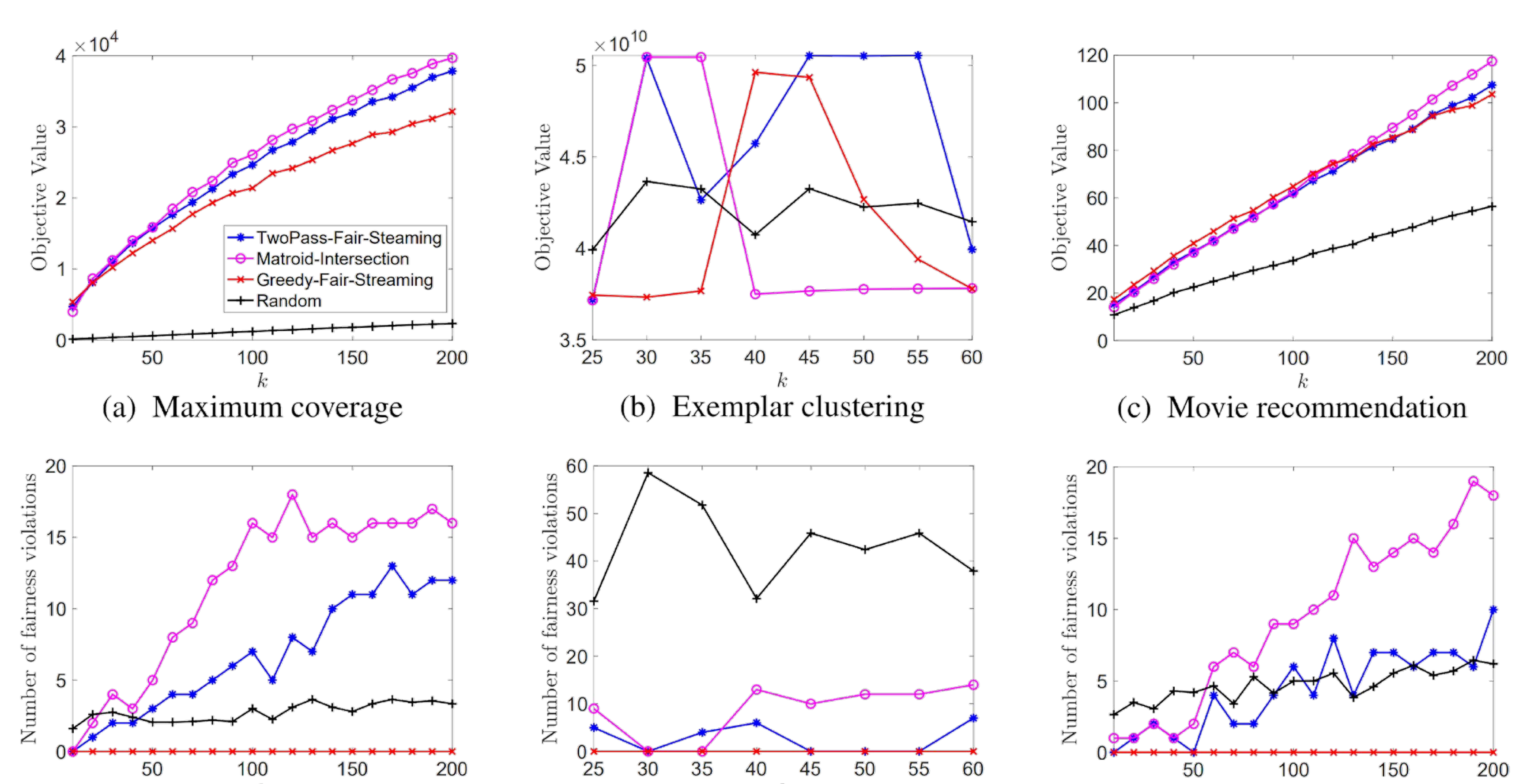
If we violate the lower bounds we can get a high solution with quadratic memory usage in two passes over the stream.

Theorem 1.3. *There exists a two-pass streaming algorithm for FMMSM that runs in polynomial time, uses $O(k \cdot C)$ memory, and outputs a set S such that (i) S is independent, (ii) it holds that $\lfloor \ell_c/2 \rfloor \le |V_c \cap S| \le u_c$ for any color $c = 1, \ldots, C$, and (iii) $f(S) \ge \mathrm{OPT}/11.656$.*

Even with more violations, it is not possible to get efficient algorithms.

Theorem 1.4. *There is no one-pass semi-streaming algorithm that determines the existence of a feasible solution for FMMSM with probability at least $2/3$, even if it is allowed to violate the fairness lower bounds by a factor of 2 and completely ignore the fairness upper bounds.*

Empirical Results



(a) Maximum coverage (b) Exemplar clustering (c) Movie recommendation

(d) Maximum coverage (e) Exemplar clustering (f) Movie recommendation

- Greedy-Fair-Streaming: a one-pass heuristic algorithm based on the first pass of our two-pass algorithm

Overview of Our Algorithm

First pass: Find any feasible solution

1. Find a solution in the matroid for each color independently.
2. Find a feasible solution in the union of these solutions.

Algorithm 1 Fair-Reservoir

- $I_c \leftarrow \emptyset$ for all $c = 1, \ldots, C$
- **for** each element e on the stream **do**
  - Let c be the color of e
  - **If** $I_c + e \in \mathcal{I}$ **then** $I_c \leftarrow I_c + e$
- Consider the partition matroid $\mathcal{I}_C$ on V defined in (1)
- $S \leftarrow$ a max-cardinality subset of $\bigcup_c I_c$ in $\mathcal{I} \cap \mathcal{I}_C$ (Lemma 2.2)
- **Return** S

Second pass: Improve the quality of the solution

1. Divide the feasible solution into two so that the lower bounds are violated by at most a factor two.
2. Extend these two sets by adding good elements to them without violating the upper bounds and the matroid constraint.
3. Return the best solution among the two.

How can we do this?

Matroid intersection

Algorithm 2 Fair-Streaming

- **Input:** Set S from Fair-Reservoir and routine $\mathcal{A}$
- $S_1 \leftarrow \emptyset$, $S_2 \leftarrow \emptyset$
- **for** e in S **do**
  - Let c be the color of e
  - **if** $|S_1 \cap V_c| < |S_2 \cap V_c|$ **then**
    - $S_1 \leftarrow S_1 + e$
  - **else**
    - $S_2 \leftarrow S_2 + e$
- Define matroids $\mathcal{I}^C, \mathcal{I}_1, \mathcal{I}_2$ as in Equations (2) and (3)
- Run two copies of $\mathcal{A}$, one for matroids $\mathcal{I}^C, \mathcal{I}_1$ and one for matroids $\mathcal{I}^C, \mathcal{I}_2$, and let S_1' and S_2' be their outputs
- **for** $i = 1, 2$ **do**
  - **for** e in S_i **do**
    - Let c be the color of e
    - **If** $|S_i' \cap V_c| < u_c$ **then** $S_i' \leftarrow S_i' + e$
- **Return** $S' = \arg\max(f(S_1'), f(S_2'))$

References

Celis, L. E., Huang, L., and Vishnoi, N. K. Multiwinner voting with fairness constraints. In IJCAI, pp. 144–151. ijcai.org, 2018.

El Halabi, M., Mitrovic, S., Norouzi-Fard, A., Tardos, J., and Tarnawski, J. Fairness in streaming submodular maximization: Algorithms and hardness. In NeurIPS, 2020.

Garg, P., Jordan, L., and Svensson, O. Semi-streaming algorithms for submodular matroid intersection. In IPCO, volume 12707 of Lecture Notes in Computer Science, pp. 208–222. Springer, 2021.

https://github.com/google-research/google-research/tree/master/fair_submodular_matroid