Simultaenous Sieves: A Deterministic Streaming Algorithm for Non-Monotone Submodular Maximization

Alan Kuhnle

October 26, 2020

Abstract

In this work, we present a combinatorial, deterministic single-pass streaming algorithm for the problem of maximizing a submodular function, not necessarily monotone, with respect to a cardinality constraint (SMCC). In the case the function is monotone, our algorithm reduces to the optimal streaming algorithm of Badanidiyuru et al. (2014). In general, our algorithm achieves ratio $\alpha/(1 + \alpha) - \varepsilon$, for any $\varepsilon > 0$, where $\alpha$ is the ratio of an offline (deterministic) algorithm for SMCC used for post-processing. Thus, if exponential computation time is allowed, our algorithm deterministically achieves nearly the optimal $1/2$ ratio. These results nearly match those of a recently proposed, randomized streaming algorithm that achieves the same ratios in expectation. For a deterministic, single-pass streaming algorithm, our algorithm achieves in polynomial time an improvement of the best approximation factor from $1/9$ of previous literature to $\approx 0.2689$.

1 Introduction

A nonnegative, set function $f : 2^U \to \mathbb{R}^+$, where ground set $U$ is of size $n$, is submodular if for all $S \subseteq T \subseteq U$, $u \in U \setminus T$: $f(T \cup \{u\}) - f(T) \leq f(S \cup \{u\}) - f(S)$. Intuitively, submodularity captures a natural diminishing returns property that arises in many machine learning applications, such as viral marketing (Kempe et al., 2003), network monitoring (Leskovec et al., 2007), video summarization (Mirzasoleiman et al., 2018), and MAP Inference for Determinantal Point Processes (Gillenwater et al., 2012). An important and well-studied NP-hard optimization problem in this context is submodular maximization subject to a cardinality constraint (SMCC): $\text{arg max}_{|S| \leq k} f(S)$, where the cardinality constraint $k$ is an input parameter and the function $f$ is submodular.

Because of the large size of the ground sets of many applications, streaming algorithms for SMCC have been developed. These algorithms store at most
$O(k \log(n))$ elements\(^1\) of the ground set and make one or more passes through the ground set. If the function $f$ is monotone, Badanidiyuru et al. (2014) introduced a single-pass streaming algorithm \texttt{SieveStream} that achieves ratio $1/2$; further, Feldman et al. (2020b) showed that to exceed the $1/2$ ratio, a single-pass algorithm must store nearly all elements of the stream. Hence, \texttt{SieveStream} is nearly optimal in approximation in polynomial time. Further, the \texttt{SieveStream} algorithm uses a simple single-threshold greedy approach, which works as follows: initially, the solution set $A = \emptyset$. When an element of the stream is received, it is added to the solution if the marginal gain to the set $A$ (i.e. $f(A \cup \{e\}) - f(A)$) exceeds a threshold of $\text{OPT}/(2k)$, where \text{OPT} is the optimal solution value. If the solution reaches size $k$, no further elements are added. To remove the requirement of knowledge of \text{OPT}, the final algorithm is modified slightly.

If the function $f$ is non-monotone, the problem is considerably more challenging. Attempts have been made to generalize \texttt{SieveStream} to non-monotone objectives; the most successful of which is the recent work of Alaluf et al. (2020), which introduced a single-pass algorithm for non-monotone SMCC based upon a randomized approach with multiple single-threshold greedy algorithms; this algorithm achieves the current state-of-the-art in approximation ratio: $0.2779$ in expectation. However, a deterministic approximation guarantee is much stronger than an expected ratio; therefore, within non-monotone submodular optimization, there has been a recent push to design deterministic algorithms (Buchbinder and Feldman, 2018; Kühnle, 2019; Feldman et al., 2020a). The best single-pass, deterministic streaming algorithm for SMCC is that of Mirzasoleiman et al. (2018), which achieves ratio $1/9$. Therefore, there is a large gap between the best deterministic ratio and the best ratio in expectation.

\textbf{Contributions} In this work, we introduce a simple, deterministic generalization of \texttt{SieveStream} to non-monotone objectives. Our algorithm simultaneously runs $\ell$ single-threshold greedy approaches, which compete with one another for incoming elements. Further, we use any $\alpha$-approximation for SMCC for a post-processing step, which consists of solving a subinstance of SMCC on the union of the $\ell$ sets obtained from the greedy approaches.

Our algorithm achieves ratio $\alpha/(1+\alpha) - \varepsilon$ by setting parameter $\ell = \lceil 1/\varepsilon \rceil$. If exponential computation is allowed, we nearly achieve the optimal ratio of $1/2$ deterministically. If the deterministic $1/e$-approximation of Buchbinder and Feldman (2018) is used, our algorithm improves the best approximation ratio in polynomial time of a deterministic single-pass algorithm from $1/9$ to $\approx 0.2689$, which nearly matches the expected ratio $0.2779$ of Alaluf et al. (2020). Further, if $f$ is monotone, one can take parameter $\ell = 1$; in this case, the post-processing is trivial to solve optimally, so $\alpha = 1$ and the resulting algorithm is identical to \texttt{SieveStream}.

\(^1\)Technically, this is the semi-streaming model.
**Related Work**  The algorithm of Alaluf et al. (2020) uses a similar strategy to achieve the same ratio of $\frac{\alpha}{1 + \alpha} - \varepsilon$ in expectation. In some sense, our algorithm can be regarded as a simplified, deterministic version of their algorithm: both use multiple copies of the single-threshold greedy algorithm and perform post-processing on the union of the solutions of the greedy algorithms to achieve ratio $\alpha/(1 + \alpha) - \varepsilon$. The analysis of our algorithm is much simpler: we avoid using the Lovasz extension of a submodular function; further, our space complexity is lower by a factor of $\Theta(\log(1/\varepsilon)/\varepsilon)$ and our time complexity is lower by a factor of $\Theta(\log(1/\varepsilon))$.

Also in Alaluf et al. (2020), a single-pass streaming algorithm with the same ratio of $\alpha/(1 + \alpha) - \varepsilon$ is introduced. This algorithm uses the multilinear extension and is deterministic if oracles to the multilinear extension and its gradient are provided; otherwise, they may be approximated with random sampling. In contrast, our algorithm does not use the multilinear extension and is deterministic on any submodular set function.

Our algorithm is also similar in some respects to the SimultaneousGreedy algorithm of Feldman et al. (2020a) and InterlaceGreedy of Kuhnle (2019), which are both non-streaming algorithms. These algorithms operate by having simultaneous greedy algorithms compete with each other; however, both of them use two greedy procedures for SMCC. In contrast, in this work we require $\ell = \lceil 1/\varepsilon \rceil$ competing, single-threshold greedy procedures.

Other streaming algorithms for non-monotone SMCC include local-search approaches (Chekuri et al., 2015; Feldman et al., 2018; Mirzasoleiman et al., 2018), of which the best expected ratio is $1/(3 + 2\sqrt{2}) \approx 0.1715$ and the best deterministic ratio is $1/9 \approx 0.1111$; the $1/9$ ratio of Mirzasoleiman et al. (2018) was the prior state-of-the-art for a deterministic single-pass algorithm for SMCC.

**Preliminaries**  We will use the following characterization of submodularity: $f$ is submodular iff for all $A \subseteq U, B \subseteq U$,

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

We will use the notation $f_A(B)$ for the marginal gain of adding set $A$ to set $B$:

$$f_A(B) = f(A \cup B) - f(B).$$

For a positive integer $m$, $[m] = \{0, 1, \ldots, m - 1\}$.

## 2 The Streaming Algorithm

Pseudocode for an idealized version of our algorithm (SimultaneousSieves) is presented in Alg. 1, which assumes knowledge of $\text{OPT}$, the optimal solution value on the instance; we discuss how to remove this assumption below. The algorithm works by running $\ell = \lceil 1/\varepsilon \rceil$ copies of a single-threshold greedy algorithm simultaneously; these greedy algorithms compete with each other in the
following way: when an element $e$ is received, the set with the highest gain for adding $e$ gets the incoming element, as long as the gain is above the threshold $\kappa = \frac{\alpha}{1+\alpha} \cdot \frac{\OPT}{k}$. At the end of the stream, post-processing is performed with any $\alpha$-approximation for SMCC; this $\alpha$-approximation is run on the union of the $\ell$ sets produced by each of the greedy algorithms. The ratio achieved by Alg. 1 is $\frac{\alpha}{1+\alpha} - \varepsilon$. If the function $f$ is monotone, one can set $\ell = 1$; then, the post-processing is trivial to perform optimally, since the resulting set is of size at most $k$ and the function is monotone. Hence, $\alpha = 1$ and SieveStream is recovered.

Using the same strategy as originally presented in Badanidiyuru et al. (2014), the assumption of knowledge of $\OPT$ can be removed by running $O(\log(k)/\varepsilon)$ parallel copies of SimultaneousSieves with different values substituted for $\OPT$. The maximum singleton seen so far in the stream is used to generate $O(\log(k)/\varepsilon)$ guesses for $\OPT$; namely:

\[ \{a(1 + \varepsilon)^m : 0 \leq m \leq \lceil \log(k)/\log(1 + \varepsilon) \rceil \}, \]

where $a$ is the maximum singleton value $f(e)$ observed thus far. As the maximum singleton value increases, new guesses are added and old ones that are too low (along with their associated algorithm) are discarded. This procedure works since a copy of Alg. 1 with a new guess would not have added any element seen thus far to any of its greedy solution sets. For full details, we refer to Badanidiyuru et al. (2014); also, see the discussion in Alaluf et al. (2020).

The resulting algorithm consists of at most $O(\log(k)/\varepsilon)$ copies of Alg. 1. Thus, the space complexity and update time stated in the theorem below follow directly.

**Theorem 1.** There is a single-pass streaming algorithm for non-negative, non-monotone submodular maximization with the following properties. At the end of the stream, the algorithm performs post-processing with an algorithm Offline with ratio $\alpha$.

- The algorithm is deterministic.
- The algorithm achieves $\frac{\alpha}{1+\alpha} - \varepsilon$ ratio.
- The algorithm uses $O(k \log(k)/\varepsilon^2)$ space.
- The update time per element received is $O(\log(k)/\varepsilon^2)$ marginal gain computations.

The rest of this section is devoted to establishing the approximation ratio of $\frac{\alpha}{1+\alpha} - \varepsilon$ for the idealized Alg. 1 with parameter $\ell$ set to $[1/\varepsilon]$. Once the stream terminates, assume that $f(A_i) < \frac{\alpha}{1+\alpha} \OPT$ for $i \in [\ell]$; otherwise, there is nothing to show. In particular, this assumption implies $|A_i| < k$ for each $i \in [\ell]$. Let $O$ be an optimal solution and let $O_1 = O \cap A$, $O_2 = O \setminus A$, where $A = \bigcup_{i \in [\ell]} A_i$ is the set defined on line 15. Let $b = |O_2|$. 


Algorithm 1: An idealized version of our streaming algorithm for non-monotone objectives, in which \( \text{OPT} \) is known.

1: \textbf{procedure} SimultaneousSieves\((f, k, \ell, \kappa)\)  
2: \textbf{Input:} oracle \( f \), cardinality constraint \( k \), integer \( \ell > 0 \), threshold \( \kappa > 0 \)  
3: \( A_i \leftarrow \emptyset \) for all \( i \in [\ell] \)  
4: \textbf{for} element \( e \) received \textbf{do}  
5: \( j \leftarrow \arg \max_{|A_i| < k} f_e(A_i) \)  
6: \textbf{if} \( f_e(A_j) \geq \kappa \) \textbf{then}  
7: \( A_j \leftarrow A_j \cup \{e\} \)  
8: \textbf{return} \{\( A_i : i \in [\ell] \)\}  
9: \textbf{procedure} PostProcess\((f, k, \varepsilon)\)  
10: \( \ell \leftarrow \lceil \frac{1}{\varepsilon} \rceil \)  
11: \( \kappa \leftarrow \frac{\alpha}{1 + \alpha} \cdot \frac{\text{OPT}}{k} \), where \( \alpha \in (0, 1] \) is the ratio of \text{Offline}  
12: \{\( A_i \} \leftarrow \text{SimultaneousSieves}(f, k, \ell, \kappa) \)  
13: \textbf{if} \( |A_i| = k \) for some \( i \in [\ell] \) \textbf{then}  
14: \textbf{return} \( A_i \)  
15: \( A \leftarrow \bigcup_{i=1}^{\ell} A_i \)  
16: \( B \leftarrow \text{Offline}(f, A, k) \)  
17: \textbf{return} arg max\{\( f(B), f(A_i) : i \in [\ell] \)\}

First, we prove a lemma that shows that the size \( b \) of \( O_2 \) influences the relationship between \( f(B) \) and \( f(A_i) \); recall that \( B \) is the solution of the offline algorithm on \( A \).

**Lemma 1.** Let \( i \in [\ell] \). Then

\[
f(A_i \cup O_1) \leq \frac{\alpha}{\alpha + 1} f(O) - bk + f(B)/\alpha
\]

**Proof.** If \( |A_i| < b \), \( |A_i \cup O_1| \leq k \) and hence is a feasible solution on the instance \((f, A, k)\). By the call to \text{Offline}, \( f(B) \geq \alpha f(A_i \cup O_1) \) and the lemma is proven. Therefore, for the rest of this proof, assume \( |A_i| \geq b \).

Let \( A' \) be composed of the first \( |A_i| - b \) elements added to \( A_i \). Notice that, by the condition to add elements to \( A_i \), we have

\[
f(A_i) \geq f(A') + bk. \tag{1}
\]

Let \( O'_1 = (O_1 \setminus A_i) \cup (A_i \setminus A') \). Observe that \( O'_1 \cap A' = \emptyset \) and \( |O'_1| \leq k \). Hence by submodularity of \( f \) and Inequality 1, we have

\[
f(A_i \cup (O_1 \setminus A_i)) = f(A' \cup O'_1) \leq f(A') + f(O'_1) \leq f(A_i) - bk + f(B)/\alpha
\]

\[
\leq \frac{\alpha}{\alpha + 1} f(O) - bk + f(B)/\alpha.
\]

\( \square \)
Next, the nature of the single-threshold greedy algorithms and the fact that $|A_i| < k$ yield the following result.

**Lemma 2.** Let $i \in [\ell]$. Then $f_{O_2}(A_i) \leq bk$.

**Proof.**

\[ f_{O_2}(A_i) \leq \sum_{o \in O_2} f_o(A_i) \leq bk, \]

by submodularity and the fact that $f_o(A_i) < \kappa$ for $o \in O_2$ since $|A_i| < k$ and $o \notin A$. \qed

Observe that $A_i \cap A_j = \emptyset$ for any $i \neq j$; this fact holds since any element is added to at most one of the greedy sets $A_i$. Because of this fact, a simple application of submodularity implies the following lemma.

**Lemma 3.** There exists $j \in [\ell]$ such that $f(O \cup A_j) \geq (1 - \varepsilon) f(O)$.

**Proof.** We will show inductively that $\sum_{j=0}^{i} f(O \cup A_i) \geq if(O) + f \left( O \cup \left( \bigcup_{j \leq i} A_j \right) \right)$, which holds trivially for $i = 0$. Suppose the statement holds for $i$ and observe that

\[
\sum_{j=0}^{i+1} f(O \cup A_i) = \sum_{j=0}^{i} f(O \cup A_i) + f(O \cup A_{i+1}) \\
\geq if(O) + f \left( O \cup \bigcup_{j \leq i} A_j \right) + f(O \cup A_{i+1}) \\
\geq (i + 1)f(O) + f \left( O \cup \bigcup_{j \leq i+1} f(A_j) \right),
\]

where the first inequality follows from the inductive assumption and the second inequality follows by an application of submodularity and the fact that $A_{i+1} \cap \bigcup_{j \leq i} A_i = \emptyset$. Therefore, $\sum_{i=1}^{\ell} f(O \cup A_i) \geq (\ell - 1)f(O)$, which implies the result of the lemma by the choice of $\ell$. \qed

**Lemma 4.** There exists $j \in [\ell]$, such that $f(O_1 \cup A_j) \geq (1 - \varepsilon)f(O) - bk$.

**Proof.** Let $j \in [\ell]$ be chosen such that $f(O_1 \cup A_j) \geq (1 - \varepsilon)f(O)$ by Lemma 3. Then

\[
f(O \cup A_j) = f_{O_2}(O_1 \cup A_j) + f(O_1 \cup A_j) \\
\leq f_{O_2}(A_j) + f(O_1 \cup A_j) \\
\leq bk + f(O_1 \cup A_j),
\]

by submodularity and Lemma 2. \qed
Finally, we combine the preceding lemmata to prove the approximation ratio of Alg. 1. Recall that we are under the assumption that all $A_i$ fail to satisfy the approximation ratio; under this assumption, we show that $B$ does.

**Lemma 5.** $f(B) \geq \left( \frac{\alpha}{1+\alpha} - \varepsilon \right) \text{OPT}$.

**Proof.** Let $j \in [\ell]$ be chosen such that $f(A_j \cup O_1) \geq (1-\varepsilon)f(O) - bk$, by Lemma 4. Then

$$(1-\varepsilon)f(O) \leq f(A_j \cup O_1) + bk$$

$$(a) \leq \frac{\alpha}{\alpha + 1} f(O) - bk + f(B)/\alpha + bk$$

$$= \frac{\alpha}{\alpha + 1} f(O) + f(B)/\alpha,$$

where (a) follows from Lemma 1. Hence

$$f(B)/\alpha \geq \left( 1 - \frac{\alpha}{\alpha + 1} - \varepsilon \right) f(O)$$

$$= \left( \frac{1}{\alpha + 1} - \varepsilon \right) f(O).$$

\[\square\]

**References**

Naor Alaluf, Alina Ene, Moran Feldman, Huy L. Nguyen, and Andrew Suh. Optimal streaming algorithms for submodular maximization with cardinality constraints. In 47th International Colloquium on Automata, Languages, and Programming (ICALP), 2020.

Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming Submodular Maximization: Massive Data Summarization on the Fly. In ACM SIGKDD Knowledge Discovery and Data Mining (KDD), pages 671–680, 2014.

Niv Buchbinder and Moran Feldman. Deterministic Algorithms for Submodular Maximization. *ACM Transactions on Algorithms*, 14(3), 2018.

Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming Algorithms for Submodular Function Maximization. In International Colloquium on Automata, Languages, and Programming (ICALP), 2015.

Moran Feldman, Amin Karbasi, and Ehsan Kazemi. Do less, Get More: Streaming Submodular Maximization with Subsampling. In Advances in Neural Information Processing Systems (NeurIPS), 2018.
Moran Feldman, Christopher Harshaw, and Amin Karbasi. Simultaneous Greedys: A Swiss Army Knife for Constrained Submodular Maximization. 2020a.

Moran Feldman, Ashkan Norouzi-Fard, Ola Svensson, and Rico Zenklusen. The One-way Communication Complexity of Submodular Maximization with Applications to Streaming and Robustness. In arXiv preprint arXiv:2003.13459, 2020b.

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Near-Optimal MAP Inference for Determinantal Point Processes. In Advances in Neural Information Processing Systems (NeurIPS), 2012.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2003.

Alan Kuhnle. Interlaced Greedy Algorithm for Maximization of Submodular Functions in Nearly Linear Time. In Advances in Neural Information Processing Systems (NeurIPS), 2019.

Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective Outbreak Detection in Networks. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2007.

Baharan Mirzasoleiman, Stefanie Jegelka, and Andreas Krause. Streaming Non-Monotone Submodular Maximization: Personalized Video Summarization on the Fly. In AAAI Conference on Artificial Intelligence, 2018.