Improved Streaming Algorithms for Maximizing Monotone Submodular Functions under a Knapsack Constraint*

Chien-Chung Huang[1,a)] Naonori Kakimura[2,b)]

Abstract: In this paper, we consider the problem of maximizing a monotone submodular function subject to a knapsack constraint in the streaming setting. In particular, the elements arrive sequentially and at any point of time, the algorithm has access only to a small fraction of the data stored in primary memory. For this problem, we propose a $(0.4 - \varepsilon)$-approximation algorithm requiring only a single pass through the data. This improves on the currently best $(0.363 - \varepsilon)$-approximation algorithm. The required memory space depends only on the size of the knapsack capacity and ε.

Keywords: Submodular functions, Streaming algorithm, Approximation algorithm

1. Introduction

A set function $f : 2^E \to \mathbb{R}_+$ on a ground set E is *submodular* if it satisfies the *diminishing marginal return property*, i.e., for any subsets $S \subseteq T \subsetneq E$ and $e \in E \setminus T$,

$$f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T).$$

A set function is *monotone* if $f(S) \leq f(T)$ for any $S \subseteq T$. Submodular functions play a fundamental role in combinatorial optimization, as they capture rank functions of matroids, edge cuts of graphs, and set coverage, just to name a few examples. Besides their theoretical interests, submodular functions have attracted much attention from the machine learning community because they can model various practical problems such as online advertising [1], [26], [37], sensor location [27], text summarization [32], [33], and maximum entropy sampling [30].

Many of the aforementioned applications can be formulated as the maximization of a monotone submodular function under a knapsack constraint. In this problem, we are given a monotone submodular function $f : 2^E \to \mathbb{R}_+$, a size function $c : E \to \mathbb{N}$, and an integer $K \in \mathbb{N}$, where $\mathbb{N}$ denotes the set of positive integers. The problem is defined as

$$\text{maximize } f(S) \quad \text{subject to } c(S) \leq K, \quad S \subseteq E, \qquad (1)$$

where we denote $c(S) = \sum_{e \in S} c(e)$ for a subset $S \subseteq E$. Note that, when $c(e) = 1$ for every item $e \in E$, the constraint coincides with a cardinality constraint. Throughout this paper, we assume that every item $e \in E$ satisfies $c(e) \leq K$ as otherwise we can simply discard it.

The problem of maximizing a monotone submodular function under a knapsack or a cardinality constraint is classical and well-studied [20], [39]. The problem is known to be NP-hard but can be approximated within the factor of $1 - e^{-1}$ (or $1 - e^{-1} - \varepsilon$); see e.g., [3], [15], [21], [28], [38], [40].

In some applications, the amount of input data is much larger than the main memory capacity of individual computers. In such a case, we need to process data in a *streaming* fashion (see e.g., [34]). That is, we consider the situation where each item in the ground set E arrives sequentially, and we are allowed to keep only a small number of the items in memory at any point. This setting effectively rules out most of the techniques in the literature, as they typically require random access to the data. In this work, we assume that the function oracle of f is available at any point of the process. Such an assumption is standard in the submodular function literature and in the context of streaming setting [2], [13], [41].

Our main contribution is to propose a single-pass $(2/5 - \varepsilon)$-approximation algorithm for the problem (1), which improves on the previous work [24], [41] (see Table 1). The space complexity is independent of the number of items in E.

Theorem 1.1 There exists a single-pass streaming $(2/5 - \varepsilon)$-approximation algorithm for the problem (1) requiring $O\left(K\varepsilon^{-4} \log^4 K\right)$ space.

2. Our Technique

Let us first describe approximation algorithms for the knapsack-constrained problem (1) in the offline setting. The simplest algorithm is a greedy algorithm, that repeatedly takes an item with maximum marginal return. The greedy algorithm admits a $(1 - 1/\sqrt{e})$-approximation, together with taking one

*A conference verson of this paper appeared in The Algorithms and Data Structures Symposium (WADS), 2019 [23].

[1] CNRS, École Normale Supérieure
[2] Department of Mathematics, Keio University
a) villars@gmail.com
b) kakimura@math.keio.ac.jp

Table 1 The knapsack-constrained problem. The algorithms [16], [38] are not for the streaming setting. See also [15], [28].

	approx. ratio	#passes	space	running time
Ours	$2/5-\varepsilon$	1	$O\left(K\varepsilon^{-4}\log^4 K\right)$	$O\left(n\varepsilon^{-4}\log^4 K\right)$
Huang *et al.* [24]	$4/11-\varepsilon$	1	$O\left(K\varepsilon^{-4}\log^4 K\right)$	$O\left(n\varepsilon^{-4}\log^4 K\right)$
Yu *et al.* [41]	$1/3-\varepsilon$	1	$O\left(K\varepsilon^{-1}\log K\right)$	$O\left(n\varepsilon^{-1}\log K\right)$
Huang *et al.* [24]	$2/5-\varepsilon$	3	$O\left(K\varepsilon^{-4}\log^4 K\right)$	$O\left(n\varepsilon^{-4}\log^4 K\right)$
Huang-Kakimura [22]	$1/2-\varepsilon$	$O\left(\varepsilon^{-1}\right)$	$O\left(K\varepsilon^{-7}\log^2 K\right)$	$O\left(n\varepsilon^{-8}\log^2 K\right)$
Ene and Nguyen [16]	$1-e^{-1}-\varepsilon$	—	—	$O\left((1/\varepsilon)^{O(1/\varepsilon^4)}n\log n\right)$
Sviridenko [38]	$1-e^{-1}$	—	—	$O\left(Kn^4\right)$

item with the maximum return, although it requires to read all the items K times. Sviridenko [38] showed that, by applying the greedy algorithm from each set of three items, we can find a $(1-1/e)$-approximate solution. Recently, such partial enumeration is replaced by a more sophisticated multi-stage guessing strategies (where fractional items are added based on the technique of multilinear extension) to improve the running time in nearly linear time [16]. However, all of them require large space and/or a large number of passes to implement.

In the streaming setting, Badanidiyuru *et al.* [2] proposed a single-pass thresholding algorithm that achieves a $(0.5-\varepsilon)$-approximation for the cardinality-constrained problem. The algorithm just takes an arriving item e when the marginal return exceeds a threshold and the feasibility is maintained. However, this strategy gives us only a $(1/3-\varepsilon)$-approximation for the knapsack-constrained problem. This drop in approximation ratio comes from the fact that, while we can freely add an item as long as our current set is of size less than K for the cardinality constraint, we cannot take a new item if its addition exceeds the capacity of the knapsack.

To overcome this issue, in [24] a branching technique is introduced, where one stops at some point of the thresholding algorithm and use a different strategy to collect subsequent items. The ratio of this branching algorithm depends on the size of the largest item o_1 in the optimal solution; when $c(o_1)$ is overly large, other strategies must be employed. Overall, the proposed approach of [24] gives a $(4/11-\varepsilon)$-approximation.

How does one improve the ratio further when $c(o_1)$ is large? One can certainly guess the size $c(o_1)$ and the f-value $f(\{o_1\})$ beforehand and in the stream pick the item of similar size and f-value. The difficulty lies in how to pick such an item that, *together with the rest of the optimal solution (excluding o_1)*, guarantees a decent f-value. Namely, we need a good substitute of o_1. In [24], a single-pass procedure, called PickOneItem, is designed to find such an item.Once equipped with such an item, it is not difficult to collect other items so as to improve the approximation ratio to $2/5-\varepsilon$. The down-side of this approach is that one needs multiple passes.

In this paper, we introduce new techniques to achieve the same ratio *without* the need to waste a pass to collect a good substitute of o_1. Depending on the relative size of o_1 and o_2 (second largest item in the optimal solution), we combine PickOneItem with the thresholding algorithm in two different ways. The first one is to perform both of them *dynamically*, that is, each time we find a candidate e for an approximation of o_1, we perform the thresholding algorithm starting from e with the current set. In contrast, the other runs both of them in a *parallel* way; we perform the thresholding algorithm and PickOneItem independently for some subset of items, and combine their results in the end. For the details, see a conference version of this paper [23].

3. Related Work

Maximizing a monotone submodular function subject to various constraints is a subject that has been extensively studied in the literature. We do not attempt to give a complete survey here and just highlight the most relevant results. Besides a knapsack constraint or a cardinality constraint mentioned above, the problem has also been studied under (multiple) matroid constraint(s), p-system constraint, multiple knapsack constraints. See [9], [11], [12], [15], [19], [28], [31] and the references therein.

In the streaming setting, Badanidiyuru *et al.* [2] proposed a single-pass $(0.5-\varepsilon)$-approximation algorithm with $O(K\varepsilon^{-1}\log K)$ space for the cardinality-constrained problem. Recently, the space complexity is improved to $O(K\varepsilon^{-1})$ [25]. Moreover, single-pass streaming algorithms have been proposed for the problem with matroid constraints [10], [18] and knapsack constraint [24], [41], and without monotonicity [13], [36]. *Multi-pass streaming algorithms*, where we are allowed to read a stream of the input multiple times, have also been studied [3], [10], [22], [24]. In particular, Chakrabarti and Kale [10] gave an $O(\varepsilon^{-3})$-pass streaming algorithms for a generalization of the maximum matching problem and the submodular maximization problem with cardinality constraint. Huang and Kakimura [22] designed an $O(\varepsilon^{-1})$-pass streaming algorithm with approximation guarantee $1/2-\varepsilon$ for the knapsack-constrained problem. Other than the streaming setting, recent applications of submodular function maximization to large data sets have motivated new directions of research on other computational models including parallel computation model such as the MapReduce model [6], [7], [29] and the adaptivity analysis [4], [5], [14], [17].

The maximum coverage problem is a special case of monotone submodular maximization under a cardinality constraint where the function is a set-covering function. For the special case, McGregor and Vu [35] and Batani *et al.* [8] gave a $(1-e^{-1}-\varepsilon)$-approximation algorithm in the multi-pass streaming setting.

Acknowledgments This work was supported by JST ERATO Grant Number JPMJER1201 and JSPS KAKENHI Grant Numbers JP18H05291 and JP17K00028, Japan.

References

[1] Alon, N., Gamzu, I. and Tennenholtz, M.: Optimizing budget allocation among channels and influencers, *Proceedings of the 21st International Conference on World Wide Web (WWW)*, pp. 381–388 (2012).

[2] Badanidiyuru, A., Mirzasoleiman, B., Karbasi, A. and Krause, A.: Streaming submodular maximization: massive data summarization on the fly, *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 671–680 (2014).

[3] Badanidiyuru, A. and Vondrák, J.: Fast algorithms for maximizing submodular functions, *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1497–1514 (2013).

[4] Balkanski, E., Rubinstein, A. and Singer, Y.: An Exponential Speedup in Parallel Running Time for Submodular Maximization without Loss in Approximation, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pp. 283–302 (online), DOI: 10.1137/1.9781611975482.19 (2019).

[5] Balkanski, E. and Singer, Y.: The Adaptive Complexity of Maximizing a Submodular Function, *Proceedings of the 50th Annual ACM Symposium on Theory of Computing*, STOC 2018, New York, NY, USA, ACM, pp. 1138–1151 (online), DOI: 10.1145/3188745.3188752 (2018).

[6] Barbosa, R. D. P., Ene, A., Nguyen, H. L. and Ward, J.: A New Framework for Distributed Submodular Maximization, *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 645–654 (online), DOI: 10.1109/FOCS.2016.74 (2016).

[7] Barbosa, R., Ene, A., Le Nguyen, H. and Ward, J.: The Power of Randomization: Distributed Submodular Maximization on Massive Datasets, *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, JMLR.org, pp. 1236–1244 (online), available from ⟨http://dl.acm.org/citation.cfm?id=3045118.3045250⟩ (2015).

[8] Bateni, M., Esfandiari, H. and Mirrokni, V.: Almost Optimal Streaming Algorithms for Coverage Problems, *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '17, New York, NY, USA, ACM, pp. 13–23 (online), DOI: 10.1145/3087556.3087585 (2017).

[9] Calinescu, G., Chekuri, C., Pál, M. and Vondrák, J.: Maximizing a Monotone Submodular Function Subject to a Matroid Constraint, *SIAM Journal on Computing*, Vol. 40, No. 6, pp. 1740–1766 (2011).

[10] Chakrabarti, A. and Kale, S.: Submodular maximization meets streaming: matchings, matroids, and more, *Mathematical Programming*, Vol. 154, No. 1-2, pp. 225–247 (2015).

[11] Chan, T.-H. H., Huang, Z., Jiang, S. H.-C., Kang, N. and Tang, Z. G.: Online Submodular Maximization with Free Disposal: Randomization Beats for Partition Matroids Online, *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1204–1223 (2017).

[12] Chan, T.-H. H., Jiang, S. H.-C., Tang, Z. G. and Wu, X.: Online Submodular Maximization Problem with Vector Packing Constraint, *Annual European Symposium on Algorithms (ESA)*, pp. 24:1–24:14 (2017).

[13] Chekuri, C., Gupta, S. and Quanrud, K.: Streaming Algorithms for Submodular Function Maximization, *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, Vol. 9134, pp. 318–330 (2015).

[14] Chekuri, C. and Quanrud, K.: Submodular Function Maximization in Parallel via the Multilinear Relaxation, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pp. 303–322 (online), DOI: 10.1137/1.9781611975482.20 (2019).

[15] Chekuri, C., Vondrák, J. and Zenklusen, R.: Submodular Function Maximization via the Multilinear Relaxation and Contention Resolution Schemes, *SIAM Journal on Computing*, Vol. 43, No. 6, pp. 1831–1879 (2014).

[16] Ene, A. and Nguyen, H. L.: A Nearly-linear Time Algorithm for Submodular Maximization with a Knapsack Constraint, *The 46th International Colloquium on Automata, Languages and Programming (ICALP 2019)*, No. to appear (2019).

[17] Ene, A. and Nguyen, H. L.: Submodular Maximization with Nearly-optimal Approximation and Adaptivity in Nearly-linear Time, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pp. 274–282 (online), DOI: 10.1137/1.9781611975482.18 (2019).

[18] Feldman, M., Karbasi, A. and Kazemi, E.: Do Less, Get More: Streaming Submodular Maximization with Subsampling, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pp. 730–740 (online), available from ⟨http://papers.nips.cc/paper/7353-do-less-get-more-streaming-submodular-maximization-with-subsampling⟩ (2018).

[19] Filmus, Y. and Ward, J.: A Tight Combinatorial Algorithm for Submodular Maximization Subject to a Matroid Constraint, *SIAM Journal on Computing*, Vol. 43, No. 2, pp. 514–542 (2014).

[20] Fisher, M. L., Nemhauser, G. L. and Wolsey, L. A.: An Analysis of Approximations for Maximizing Submodular Set Functions I, *Mathematical Programming*, pp. 265–294 (1978).

[21] Fisher, M. L., Nemhauser, G. L. and Wolsey, L. A.: An Analysis of Approximations for Maximizing Submodular Set Functions II, *Mathematical Programming Study*, Vol. 8, pp. 73–87 (1978).

[22] Huang, C. and Kakimura, N.: Multi-Pass Streaming Algorithms for Monotone Submodular Function Maximization, *CoRR*, Vol. abs/1802.06212 (online), available from ⟨http://arxiv.org/abs/1802.06212⟩ (2018).

[23] Huang, C.-C. and Kakimura, N.: Improved Streaming Algorithms for Maximizing Monotone Submodular Functions Under a Knapsack Constraint, *Algorithms and Data Structures* (Friggstad, Z., Sack, J.-R. and Salavatipour, M. R., eds.), Cham, Springer International Publishing, pp. 438–451 (2019).

[24] Huang, C.-C., Kakimura, N. and Yoshida, Y.: Streaming Algorithms for Maximizing Monotone Submodular Functions under a Knapsack Constraint, *The 20th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems(APPROX2017)* (2017).

[25] Kazemi, E., Mitrovic, M., Zadimoghaddam, M., Lattanzi, S. and Karbasi, A.: Submodular Streaming in All Its Glory: Tight Approximation, Minimum Memory and Low Adaptive Complexity, *International Conference on Machine Learning (ICML2019)*, pp. 3311–3320 (2019).

[26] Kempe, D., Kleinberg, J. and Tardos, É.: Maximizing the spread of influence through a social network, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 137–146 (2003).

[27] Krause, A., Singh, A. P. and Guestrin, C.: Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies, *Journal of Machine Learning Research*, Vol. 9, pp. 235–284 (2008).

[28] Kulik, A., Shachnai, H. and Tamir, T.: Maximizing Submodular Set Functions Subject to Multiple Linear Constraints, *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 545–554 (2013).

[29] Kumar, R., Moseley, B., Vassilvitskii, S. and Vattani, A.: Fast Greedy Algorithms in MapReduce and Streaming, *ACM Trans. Parallel Comput.*, Vol. 2, No. 3, pp. 14:1–14:22 (online), DOI: 10.1145/2809814 (2015).

[30] Lee, J.: *Maximum Entropy Sampling*, Encyclopedia of Environmetrics, Vol. 3, pp. 1229–1234, John Wiley & Sons, Ltd. (2006).

[31] Lee, J., Sviridenko, M. and Vondrák, J.: Submodular Maximization over Multiple Matroids via Generalized Exchange Properties., *Mathematics of Operations Research*, Vol. 35, No. 4, pp. 795–806 (2010).

[32] Lin, H. and Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions, *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 912–920 (2010).

[33] Lin, H. and Bilmes, J.: A class of submodular functions for document summarization, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pp. 510–520 (2011).

[34] McGregor, A.: Graph Stream Algorithms: A Survey, *SIGMOD Rec.*, Vol. 43, No. 1, pp. 9–20 (online), DOI: 10.1145/2627692.2627694 (2014).

[35] McGregor, A. and Vu, H. T.: Better Streaming Algorithms for the Maximum Coverage Problem, *International Conference on Database Theory (ICDT)* (2017).

[36] Mirzasoleiman, B., Jegelka, S. and Krause, A.: Streaming Non-monotone Submodular Maximization: Personalized Video Summarization on the Fly, *Proc. Conference on Artificial Intelligence (AAAI)* (2018).

[37] Soma, T., Kakimura, N., Inaba, K. and Kawarabayashi, K.: Optimal Budget Allocation: Theoretical Guarantee and Efficient Algorithm, *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pp. 351–359 (2014).

[38] Sviridenko, M.: A note on maximizing a submodular set function subject to a knapsack constraint, *Operations Research Letters*, Vol. 32, No. 1, pp. 41–43 (2004).

[39] Wolsey, L.: Maximising real-valued submodular functions: primal and dual heuristics for location problems, *Mathematics of Operations Research* (1982).

[40] Yoshida, Y.: Maximizing a Monotone Submodular Function with a

Bounded Curvature under a Knapsack Constraint, *SIAM Journal on Discrete Mathematics*, Vol. 33, No. 3, pp. 1452–1471 (online), DOI: 10.1137/16M1107644 (2019).

[41] Yu, Q., Xu, E. L. and Cui, S.: Streaming Algorithms for News and Scientific Literature Recommendation: Submodular Maximization with a d-Knapsack Constraint, *IEEE Global Conference on Signal and Information Processing* (2016).