SUBMODULAR MAXIMIZATION WITH MULTI-KNAPSACK CONSTRAINTS AND ITS APPLICATIONS IN SCIENTIFIC LITERATURE RECOMMENDATIONS

Qilian Yu, *Student Member, IEEE*, Easton Li Xu, *Member, IEEE,* and Shuguang Cui, *Fellow, IEEE*

ABSTRACT

Submodular maximization problems belong to the family of combinatorial optimization problems and enjoy wide applications. In this paper, we focus on the problem of maximizing a monotone submodular function subject to a d-knapsack constraint, for which we propose a streaming algorithm that achieves a $\left(\frac{1}{1+2d}-\epsilon\right)$-approximation of the optimal value, while it only needs one single pass through the dataset without storing all the data in the memory. In our experiments, we extensively evaluate the effectiveness of our proposed algorithm via an application in scientific literature recommendation. It is observed that the proposed streaming algorithm achieves both execution speedup and memory saving by several orders of magnitude, compared with existing approaches.

Index Terms— Submodular Optimization, Streaming Algorithm, Scientific Literature Recommendation

1. INTRODUCTION

As our society enters the big data era, the main problem that data scientists are facing is how to process the unprecedented large datasets. Besides, data sources are heterogenous, comprising documents, images, sounds, and videos. Such challenges require the data processing algorithms to be more computationally efficient. The concept of submodularity plays an important role in pursuing efficient solutions for combinatorial optimization, since it has rich theoretical and practical features. Hence submodular optimization has been adopted to preprocess massive data in order to reduce the computational complexity [1–3].

Although maximizing a submodular function under a cardinality constraint is an NP-hard problem, a greedy algorithm developed in [4] achieving a $(1-e^{-1})$-approximation of the optimal solution can be easily applied for various applications. However, large-scale problems prevent the greedy algorithm from being adequate due to practical computation resource and memory limitations. And it is possible that the number of data samples grows rapidly such that the main memory is not able to read all of them simultaneously.

Under the scenarios discussed above, processing data in a streaming fashion becomes a necessity, where at any time point, the streaming algorithm needs to store just a small portion of data into the main memory, and produces the solution right at the end of data stream with limited computation resource. In [5], the authors introduced a streaming algorithm to maximize a submodular function under a constraint, where the cardinality constraint is just a special case of a d-knapsack constraint [6] with each weight being one. When each element has multiple weights or there are more than one knapsack constraints, the algorithm in [5] is no longer applicable.

In this paper, we develop a new streaming algorithm to maximize a monotone submodular function, subject to a general d-knapsack constraint. It requires only one single pass through the data, and produces a $\left(\frac{1}{1+2d}-\epsilon\right)$-approximation of the optimal solution, for any $\epsilon > 0$. In addition, the algorithm only requires $O\left(\frac{b\log b}{d\epsilon}\right)$ memory (independent of the dataset size) and $O\left(\frac{\log b}{\epsilon}\right)$ computation per element with b being the standardized d-knapsack capacity. To our knowledge, it is the first streaming algorithm that provides a constant-factor approximation guarantee with only monotone submodularity assumed.

The rest of this paper is organized as follows. In Section 2 we introduce the formulation and related existing results. In Section 3 we describe the proposed algorithms. In Section 4 we present an application in scientific literature recommendations. We draw the conclusions in Section 5.

2. FORMULATION AND MAIN RESULTS

2.1. Problem Formulation

Let $V=\{1,2,\ldots,n\}$ be the ground set and $f: 2^V \to [0,\infty)$ be a nonnegative set function on the subsets of V. For any subset S of V, we denote the characteristic vector of S by $\mathbf{x}_S = (x_{S,1}, x_{S,2}, \ldots, x_{S,n})$, where for $1 \le j \le n$, $x_{S,j} = 1$, if $j \in S$; $x_{S,j} = 0$, otherwise. For $S \subseteq V$ and $r \in V$, the marginal gain of f with respect to S and r is defined to be $\Delta_f(r|S) \triangleq f(S\cup\{r\}) - f(S)$, which quantifies the increase in $f(S)$ when r is added into subset S. A function f is submodular if it satisfies that $\Delta_f(r|B) \le \Delta_f(r|A)$, for

The work of Q. Yu, E. L. Xu, and S. Cui was supported in part by DoD with grant HDTRA1-13-1-0029, by grant NSFC-61328102/61629101, and by NSF with grants AST-1547436, ECCS-1508051, CNS-1343155, ECCS-1305979, and CNS-1265227.

Q. Yu and S. Cui are with the Department of Electrical and Computer Engineering, University of California, Davis, CA 95616, USA (emails: {dryu, sgcui}@ucdavis.edu), and E. L. Xu is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA (email: eastonlixu@tamu.edu).

any $A \subseteq B \subseteq V$ and $r \in V \setminus B$. Also, f is said to be a monotone function, if for any $S \subseteq V$ and $r \in V$, $\Delta_f(r|S) \geq 0$. In Section 3, we will discuss the case when the submodular function is independent [7] of the ground set V. When the value of $f(S)$ depends on the whole ground set V, and f is *additively decomposable* [7], our proposed algorithm is still useful by randomly choosing a small subset $\widetilde{V}$ of V, and use $f_{\widetilde{V}}(S) \triangleq \frac{1}{|\widetilde{V}|}\sum_{i\in\widetilde{V}} f_i(S)$ instead of f in the algorithm.

Next, we introduce the d-knapsack constraint. Let $\mathbf{b} = (b_1, b_2, \ldots, b_d)^T$ be a d-dimensional budget vector, where for $1 \leq i \leq d$, $b_i > 0$ is the budget corresponding to the i-th resource. Let $C = (c_{i,j})$ denote a $d \times n$ matrix, whose (i,j)-th entry $c_{i,j} > 0$ is the weight of the element $j \in V$ with respect to the i-th knapsack resource constraint. Then the d-knapsack constraint can be expressed by $C\mathbf{x}_S \leq \mathbf{b}$. The problem for maximizing a monotone submodular function f subject to a d-knapsack constraint can be formulated as

$$\begin{aligned} \underset{S\subseteq V}{\text{maximize}} \quad & f(S) \\ \text{subject to} \quad & C\mathbf{x}_S \leq \mathbf{b}. \end{aligned} \tag{1}$$

We aim to **MA**ximize a monotone **S**ubmodular set function subject to a d-**K**napsack constraint, which is called d-**MASK** for short. Without loss of generality, for $1 \leq i \leq d, 1 \leq j \leq n$, we assume that $c_{i,j} \leq b_i$. That is, no entry in C has a larger weight than the corresponding knapsack budget, since otherwise the corresponding element is never selected into S.

For the sake of simplicity, we standardize Problem (1). Let $b \triangleq \max_{1\leq i\leq d} b_i$ and $c' \triangleq \min_{1\leq i\leq d, 1\leq j\leq n} bc_{i,j}/b_i$. We replace each $c_{i,j}$ with $bc_{i,j}/b_ic'$ and b_i with b/c'. Now $c_{i,j} \geq 1$ and $b_i = b$, for $1 \leq i \leq d$, $1 \leq j \leq n$. The standardized problem has the same optimal solution as Problem (1). In the rest of the paper, we only consider the standardized version of the d-MASK problem.

2.2. Related Work and Main Results

Submodular optimization has been regarded as a powerful tool for combinatorial massive data mining and machine learning, for which a streaming algorithm processes the dataset piece by piece and then produces an approximate solution at the end of the data stream. This makes it quite suitable to process a massive dataset in many applications.

When $d = 1$ and all entries of C are ones, Problem (1) is equivalent to maximizing a monotone submodular function under a cardinality constraint. This optimization problem has been proved to be NP-hard [4], the greedy algorithm [4] is proposed and produces a $(1 - e^{-1})$-approximation guarantee with $O(kn)$ computation complexity, where k is the maximum number of elements that the solution set can include. Recently, some accelerated algorithms were proposed in [8, 9]. Unfortunately, neither of them can be applied to the case when the size of the dataset is over the capacity of the main memory. Further, a streaming algorithm was developed in [5] with a $(1/2 - \epsilon)$-approximation of the optimal value, for any $\epsilon > 0$. This streaming algorithm does not require the full access to the dataset, and needs only one pass through the dataset. Thus it provides a practical way to process a large dataset on the fly with a low memory requirement, but not applicable under a general d-knapsack constraint.

Further, the authors in [10] dealt with the case when $d = 1$ and each entry of C can take any positive values. This problem is called a budgeted submodular maximization problem and is proved NP-hard, and the authors in [11] suggested a greedy algorithm, producing a $(1 - e^{-1})$-approximation of the optimal value with $O(n^5)$ computation complexity. However, the high computation cost prevents this algorithm from being widely used in practice. Hence some modified algorithms [10, 12] have been developed.

The considered d-MASK problem is a generalization of the above problems to maximize a submodular function under more than one budgeted constraints. A framework was proposed in [6] for maximizing a submodular function subject to a d-knapsack constraint, which yields a $(1 - e^{-1} - \epsilon)$-approximation for any $\epsilon > 0$. However, it is hard to implement this algorithm, since it involves some high-order terms with respect to the number of budgets, making it inappropriate for processing large datasets [13]. Later, an accelerated algorithm providing an $\Omega(1/d)$-approximation was developed in [14]. However, this algorithm needs an $O(\log n)$ blowup in communication complexity among various parts. As observed in [15], such a blowup decreases its applicability in practice. Note that the authors in [14] mentioned that the MapReduce method with an $\Omega(1/d)$-approximation can be extended to execute in a streaming fashion, but did not provide any concrete algorithms and the associated analysis.

To our best knowledge, this paper is the first to propose an efficient streaming algorithm for maximizing a monotone submodular function under a d-knapsack constraint, with 1) a constant-factor approximation guarantee, 2) no assumption on full access to the dataset, 3) execution of a single pass, 4) $O(b \log b)$ memory requirement, 5) $O(\log b)$ computation complexity per element, and 6) only assumption on monotonicity and submodularity of the objective function. In the following, we describe the proposed algorithm in details.

3. STREAMING ALGORITHMS FOR MAXIMIZING MONOTONE SUBMODULAR FUNCTIONS

We develop our one-pass streaming algorithm as follows:

Algorithm 1 d-KNAPSACK-STREAMING

1: $m := 0$.
2: $Q := \{[1 + (1 + 2d)\varepsilon]^l | l \in \mathbb{Z}\}$.
3: **for** $v \in Q$
4: $\quad S_v := \emptyset$.
5: $\quad$ **for** $i := 1$ **to** d

Table 1: Comparison of approximation guarantees and computation costs

	Best Performance Known Algorithms		Proposed Streaming Algorithms	
	Approx. Factor	Comput. Cost	Approx. Factor	Comput. Cost
1-Knapsack Constraint	$1-e^{-1}$	$O(n^5)$	$1/(1+2d)-\epsilon$	$O(n\log b/\epsilon)$
d-Knapsack Constraint	$1-e^{-1}-\epsilon$	Polynomial		

$m := \max\{m, f(\{j\})/c_{i,j}\}$.
end for
$Q := \{[1+(1+2d)\varepsilon]^l \big| l \in \mathbb{Z}, \frac{m}{1+(1+2d)\epsilon} \le [1+(1+2d)\varepsilon]^l \le 2bm\}$.
for $j := 1$ **to** n
if $c_{i,j} \ge \frac{b}{2}$ and $\frac{f(\{j\})}{c_{i,j}} \ge \frac{2v}{b(1+2d)}$ for some $i \in [1,d]$ **then**
$S_v := \{j\}$.
break
end if
if $\sum_{l\in S\cup\{j\}} c_{i,l} \le b$ and $\frac{\Delta_f(j|S)}{c_{i,j}} \ge \frac{2v}{b(1+2d)}$ for all $i \in [1,d]$ **then**
$S_v := S_v \cup \{j\}$.
end if
end for
end for
$S := \underset{S_v, v\in Q}{\operatorname{argmax}} f(S_v)$.
return S.

Choose ϵ with $0 < \epsilon < \frac{1}{1+2d}$. In the algorithm, we set Q to be $\{[1+(1+2d)\epsilon]^l \big| l \in \mathbb{Z}, \frac{m}{1+(1+2d)\epsilon} \le [1+(1+2d)\epsilon]^l \le 2bm\}$, where m holds the current maximum marginal value per weight of all single element. It can be proved that there exists at least some $v \in Q$ such that $[1-(1+2d)\epsilon]\text{OPT} \le v \le \text{OPT}$. Here OPT is the optimal value of the optimization problem 1. For $v \in Q$, at the beginning of the algorithm, candidate solution set S_v is set to be an empty set. When the algorithm finds an element a such that

$$c_{i,j} \ge \frac{b}{2} \text{ and } \frac{f(\{j\})}{c_{i,j}} \ge \frac{2v}{b(1+2d)} \text{ for some } i \in [1,d],$$

$\{a\}$ is simply output as S_v. When V does not contain such elements, during the data streaming, an element j is added to the solution set S_v if 1) the marginal value per weight for each knapsack constraint $\Delta_f(j|S)/c_{i,j}$ is at least $2v/b(1+2d)$ for $1 \le i \le d$, and 2) the overall d-knapsack constraint is still satisfied. Finally, after the algorithm finishes one pass through the dataset, the algorithm outputs S_v with the largest function value. We compare the approximation guarantees and computation costs of the algorithms [4, 6, 11] against our proposed algorithm in Table 1, which shows our streaming algorithm achieves both execution speedup by several orders of magnitude, compared with existing approaches. The following theorem shows the property of the output S of Algorithm 1, whose detailed proof can be found in [16].

Theorem 1. *Algorithm 1 has the following properties:*

- *It outputs S that satisfies that $f(S) \ge \left(\frac{1}{1+2d}-\epsilon\right)$ OPT;*
- *It goes one pass over the dataset, stores at most $O\left(\frac{b\log b}{d\epsilon}\right)$ elements, and has $O\left(\frac{\log b}{\epsilon}\right)$ computation complexity per element.*

To evaluate the performance of Algorithm 1, we need to compare the function values obtained by our streaming algorithm against OPT, by calculating their relative difference. Since OPT is unknown, we could use an upper bound of OPT to evaluate the performance of the proposed algorithms.

By Theorem 1, we obtain $\text{OPT} \le \frac{1+2d}{1-(1+2d)\epsilon} f(S)$. Then $\frac{1+2d}{1-(1+2d)\epsilon} f(S)$ is an upper bound of the optimal value to the d-MASK problem. In most of cases, this bound is not tight enough. So we provide a much tighter bound derived by the submodularity of f in the following.

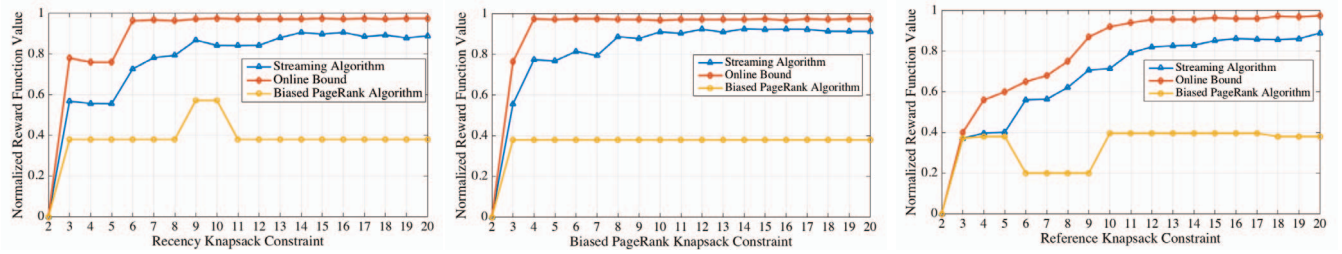
Theorem 2. *Consider a subset $S \subseteq V$. For $1 \le i \le d$, let $r_{i,s} = \Delta_f(s|S)/c_{i,s}$, and $s_{i,1}, \ldots, s_{i,|V\setminus S|}$ be the sequence such that $r_{i,s_{i,1}} \ge r_{i,s_{i,2}} \ge \cdots \ge r_{i,s_{i,|V\setminus S|}}$. Let k_i be the integer such that $\sum_{j=1}^{k_i-1} c_{i,s_{i,j}} \le b$ and $\sum_{j=1}^{k_i} c_{i,s_{i,j}} > b$. And let $\lambda_i = \left(b - \sum_{j=1}^{k_i-1} c_{i,s_{i,j}}\right) \Big/ c_{i,s_{i,k_i}}$. Then we have*

$$\text{OPT} \le f(S) + \min_{1\le i\le d} \left[\sum_{j=1}^{k_i-1} \Delta_f(s_{i,j}|S) + \lambda_i \Delta_f(s_{i,k_i}|S)\right].$$

4. APPLICATIONS

In this section, we discuss a real-world application for Algorithm 1 in scientific literature recommendation. Nowadays, the researchers have to face an enormous amount of articles, where they have to filter the massive scientific literatures and pick the most useful ones. A common approach to locate the targeted literatures is based on the so-called citation networks [17]. The authors in [17] mapped a citation network onto a rating matrix to filter research papers. In [18], an algorithm utilizing the random-walker properties was proposed. It transforms a citation matrix into a probability transition matrix and outputs the entries with the highest biased PageRank scores.

We propose a new scientific literature recommendation system based on the citation networks. Consider a directed



(a) (b) (c)

Fig. 1: a) Optimal Function Values corresponding to Different Recency Constraints, b) Optimal Function Values corresponding to Different Biased PageRank Constraints, c) Optimal Function Values corresponding to Different Reference Knapsack Constraints

acyclic graph $G = (V, E)$ with $V = \{1, 2, \ldots, n\}$, where each vertex in V represents an article. For $i, j \in V$, arc $(i, j) \in E$ if and only if paper i cites paper j. The information initiates from a set of vertices (source papers), and then spreads over the reverse directions of the arcs in the network. Let A be the collection of the source papers. Our target is to select a subset S out of V to quickly detect the information spreading of A. This problem can be formulated as a monotone submodular maximization under a 3-knapsack constraint.[1]

Observe that the papers in A transfer their influence through the citation network, but this influence becomes less as it spreads through more hops. Let $T(s, a)$ be the length of the shortest directed path from s to a. Then the shortest path length from any vertex in S to a is defined as $T(S, a) \triangleq \min_{s \in S} T(s, a)$. Let $W(a)$ be a pre-assigned weight to each vertex $a \in A$ such that $\sum_{a \in A} W(a) = 1$. Then our goal is to minimize the expected penalty $\pi(S) \triangleq \sum_{a \in A} W(a) \min\{T(S, a), T_{\max}\}$, or maximize the expected penalty reduction $R(S) \triangleq \sum_{a \in A} W(a)[T_{\max} - T(S, a)]^+$, which is a monotone submodular function. Here $[x]^+ \triangleq \max\{x, 0\}$ and $T_{\max}$ is a given maximum penalty.

We construct three constraints in (1) from the aspects of recency, biased PageRank score, and reference number respectively. The first aspect is from the fact that readers prefer to read the recently published papers. Let $c_{1,j}$ be the time difference between the publishing date of paper j and the current date, and b_1 be the corresponding limit. For the second aspect, we use the biased PageRank score introduced in [18]. It is a measure of the significance of each paper, not only involving the propagation and attenuation properties of the network, but also taking the set of source vertices into account. Let $\rho(j)$ be the biased PageRank score of article j. We further choose a function $\xi(x) \triangleq \frac{2+x}{1+x}$ to map the PageRank score onto $(1, 2]$. Then paper j with the smaller value $c_{2,j} \triangleq \xi(\rho(j))$ is more valuable for the researchers. Also we set b_2 to be corresponding budget. Thirdly, we assume that more references listed in the paper, more time the reader spends on picking the information. Then we set $c_{3,j}$ to be the number of references in paper j and b_3 be the budget of the total number of references.

To evaluate the performance of Algorithm 1, we utilize a dataset collected in [19], which includes more than 20,000 papers in the Association of Computational Linguistics. We compare the function values obtained by Algorithm 1 and the PageRank algorithm proposed in [18].

We perform the sensitive analysis over different knapsack constraints. With the other two constraints fixed, we change the value of the budget corresponding to the recency, biased PageRank score or reference number, respectively. Here we randomly select five nodes as the source papers. We set $T_{\max} = 50$ and $W(a) = 0.2$ for each source paper a. The results for the optimal objective values are shown in Fig. 1(a) (with fixed $b_2 = 10$, $b_3 = 20$), Fig. 1(b) (with fixed $b_1 = 20$, $b_3 = 20$) and Fig. 1(c) (with fixed $b_1 = 20$, $b_2 = 10$), respectively. It can be observed that the relative difference is around 10% between the function values obtained by our streaming algorithm (blue lines) and the corresponding online bounds (red lines). Also, we find that our algorithm highly outperforms the biased PageRank algorithm. Although the biased PageRank algorithm suggests the papers with high biased PageRank scores, most of the suggested papers have very long distances from the set of source articles, which leads to a very low objective function value.

5. CONCLUSIONS

In this paper, we proposed a streaming algorithm to maximize a monotone submodular function under a d-knapsack constraint. It leads to a $\left(\frac{1}{1+2d} - \epsilon\right)$ approximation of the optimal value, and requires only a single pass through the dataset and a small memory size. It achieves a major fraction of the utility function value obtained by the greedy algorithm with a much lower computation cost, which makes it very practically implementable. Our algorithm provides a more efficient way to solve the related combinatorial optimization problems, which could find many good applications, like scientific literature recommendations as shown in the paper.

[1]The reason why we set $d = 3$ will be explained later in this section; based on the different usages, the number of knapsack constraints and the corresponding budgets can be changed accordingly.

6. REFERENCES

[1] Y. Liu, K. Wei, K. Kirchhoff, Y. Song, and J. Bilmes, "Submodular feature selection for high-dimensional acoustic score spaces," in *Proc. 2013 IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, BC, May 2013, pp. 7184–7188.

[2] H. Lin and J. Bilmes, "How to select a good training-data subset for transcription: Submodular active selection for sequences," Tech. Rep., Brighton, UK, Sept. 2009.

[3] S. Chakraborty, O. Tickoo, and R. Iyer, "Adaptive keyframe selection for video summarization," in *Proc. 2015 IEEE Winter Conf. Applicat. Comput. Vision*, Waikoloa, HI, Jan. 2015, pp. 702–709.

[4] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions–I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, Dec. 1978.

[5] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause, "Streaming submodular maximization: Massive data summarization on the fly," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, Aug. 2014, pp. 671–680.

[6] A. Kulik, H. Shachnai, and T. Tamir, "Maximizing submodular set functions subject to multiple linear constraints," in *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algor.*, New York, NY, Jan. 2009, pp. 545–554.

[7] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause, "Distributed submodular maximization: Identifying representative elements in massive data," Dec. 2013, pp. 2049–2057.

[8] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, Jan. 2015, pp. 1812–1818.

[9] A. Badanidiyuru, Ashwinkumar, and J. Vondrák, "Fast algorithms for maximizing submodular functions," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algor.*, Portland, OR, Oct. 2014, pp. 1497–1514.

[10] H. Lin and J. Bilmes, "Multi-document summarization via budgeted maximization of submodular functions," in *Proc. 2010 NAACL HLT*, Los Angeles, CA, June 2010, pp. 912–920.

[11] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Oper. Res. Lett*, vol. 32, pp. 41–43, Jan. 2004.

[12] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, San Jose, CA, Aug. 2007, pp. 420–429.

[13] G. Papachristoudis, *Theoretical Guarantees and Complexity Reduction in Information Planning*, Ph.D. thesis, MIT, June 2015.

[14] R. Kumar, B. Moseley, S. Vassilvitskii, and A. Vattani, "Fast greedy algorithms in mapreduce and streaming," in *Proc. 25th Annu. ACM Symp. Parallelism Algor. Archit.*, Montreal, QC, June 2013, pp. 1–10.

[15] R. Kiveris, S. Lattanzi, V. Mirrokni, V. Rastogi, and S. Vassilvitskii, "Connected components in mapreduce and beyond," in *Proc. ACM Symp. Cloud Comput.*, Seattle, WA, June 2014, pp. 1–13.

[16] Q. Yu, E. L. Xu, and S. Cui, "Streaming algorithms for news and scientific literature recommendation: Submodular maximization with a d-knapsack constraint," *ArXiv:1603.05614*.

[17] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. Rashid, J. A. Konstan, and J. Riedl, "On the recommending of citations for research papers," in *Proc. 2002 ACM Conf. Comput. Support. Coop. Work*, New Orleans, LA, Nov. 2002, pp. 116–125.

[18] M. Gori and A. Pucci, "Research paper recommender systems: A random-walk based approach," in *Proc. 2006 IEEE/WIC/ACM Int. Conf. Web Intell.*, Hong Kong, Dec. 2006, pp. 778–781.

[19] M. T. Joseph and D. R. Radev, "Citation analysis, centrality, and the ACL anthology," Tech. Rep., CSE-TR-535-07, Oct. 2007.