Maximum Coverage in the Data Stream Model: Parameterized and Generalized

Andrew McGregor
University of Massachusetts, Amherst

David Tench
Stony Brook University

Hoa T. Vu
San Diego State University

Abstract

We present algorithms for the `Max Coverage` and `Max Unique Coverage` problems in the data stream model. The input to both problems are m subsets of a universe of size n and a value $k \in [m]$. In `Max Coverage`, the problem is to find a collection of at most k sets such that the number of elements covered by at least one set is maximized. In `Max Unique Coverage`, the problem is to find a collection of at most k sets such that the number of elements covered by exactly one set is maximized. These problems are closely related to a range of graph problems including matching, partial vertex cover, and capacitated maximum cut. In the data stream model, we assume k is given and the sets are revealed online. Our goal is to design single-pass algorithms that use space that is sublinear in the input size. Our main algorithmic results are:

- If the sets have size at most d, there exist single-pass algorithms using $O(d^{d+1}k^d)$ space that solve both problems exactly. This is optimal up to polylogarithmic factors for constant d.
- If each element appears in at most r sets, we present single pass algorithms using $\tilde{O}(k^2r/\epsilon^3)$ space that return a $1+\epsilon$ approximation in the case of `Max Coverage`. We also present a single-pass algorithm using slightly more memory, i.e., $\tilde{O}(k^3r/\epsilon^4)$ space, that $1+\epsilon$ approximates `Max Unique Coverage`.

In contrast to the above results, when d and r are arbitrary, any constant pass $1+\epsilon$ approximation algorithm for either problem requires $\Omega(\epsilon^{-2}m)$ space but a single pass $O(\epsilon^{-2}mk)$ space algorithm exists. In fact any constant-pass algorithm with an approximation better than $e/(e-1)$ and $e^{1-1/k}$ for `Max Coverage` and `Max Unique Coverage` respectively requires $\Omega(m/k^2)$ space when d and r are unrestricted. En route, we also obtain an algorithm for a parameterized version of the streaming `Set Cover` problem.

2012 ACM Subject Classification Theory of computation → Sketching and sampling; Theory of computation → Approximation algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Data streams, maximum coverage, maximum unique coverage, set cover

Funding This work was partially supported by NSF grants CCF-1934846, CCF-1908849, and CCF-1637536.

1 Introduction

Problem Description. We consider the `Max Coverage` and `Max Unique Coverage` problems in the data stream model. The input to both problems are m subsets of a universe of size n and a value $k \in [m]$. In `Max Coverage`, the problem is to find a collection of at most k sets such that the number of elements covered by at least one set is maximized. In `Max Unique Coverage`, the problem is to find a collection of at most k sets such that the number of elements covered by exactly one set is maximized. In the data stream model, we assume k is provided but that the sets are revealed online and our goal is to design single-pass algorithms that use space that is sub-linear in the input size.

Max Coverage is a classic NP-Hard problem that has a wide range of applications including facility and sensor allocation [52], information retrieval [5], influence maximization in marketing strategy design [48], and the blog monitoring problem [64]. It is well-known that the greedy algorithm, which greedily picks the set that covers the most number of uncovered elements, is a $e/(e-1)$ approximation and that unless P = NP, this approximation factor is the best possible in polynomial time [30].

Max Unique Coverage was first studied in the offline setting by Demaine et al. [25]. A motivating application for this problem was in the design of wireless networks where we want to place base stations that cover mobile clients. Each station could cover multiple clients but unless a client is covered by a unique station the client would experience too much interference. Demaine et al. [25] gave a polynomial time $O(\log k)$ approximation. Furthermore, they showed that Max Unique Coverage is hard to approximate within a factor $O(\log^{\sigma} n)$ for some constant σ under reasonable complexity assumptions. Erlebach and van Leeuwen [29] and Ito et al. [40] considered a geometric variant of the problem and Misra et al. [62] considered the parameterized complexity of the problem. This problem is also closely related to Minimum Membership Set Cover where one has to cover every element and minimizes the maximum overlap on any element [26, 53].

In the streaming set model, Max Coverage and the related Set Cover problem[1] have both received a significant amount of attention [7, 15, 27, 36, 38, 39, 61, 64]. The most relevant result is a single-pass $2+\epsilon$ approximation using $\tilde{O}(k\epsilon^{-3})$ space [8,61] although better approximation is possible in a similar amount of space if multiple passes are permitted [61] or if the stream is randomly ordered [2, 63]. In this paper, we almost exclusively consider single-pass algorithms where the sets arrive in an arbitrary order.

The unique coverage problem has not been studied in the data stream model although it, and Max Coverage, are closely related to various graph problems that have been studied.

Relationship to Graph Streaming. There are two main variants of the graph stream model. In the *arbitrary order model*, the stream consists of the edges of the graph in arbitrary order. In the *adjacency list model*, all edges that include the same node are grouped together. Both models generalize naturally to hypergraphs where each edge could consists of more than two nodes. The arbitary order model has been more heavily studied than the adjacency list model but there has still been a significant amount of work in the latter model [6,7,11,36,42,50,57–59]. For further details, see a recent survey on work on the graph stream model [56].

To explore the relationship between Max Coverage and Max Unique Coverage and various graph stream problems, it makes sense to introduce to additional parameters beyond m (the number of sets) and n (the size of the universe). Specifically, throughout the paper we let d denote the maximum cardinality of a set in the input and let r denote the maximum multiplicity of an element in the universe where the *multiplicity* is the number of sets an element appears.[2] Then an input to Max Coverage and Max Unique Coverage can define a (hyper)graph in one of the following two natural ways:

1. *First Interpretation:* A sequence of (hyper-)edges on a graph with n nodes of maximum degree r (where the degree of a node v corresponds to how many hyperedges include that node) and m hyperedges where each hyperedge has size at most d. In the case where every set has size $d = 2$, the hypergraph is an *ordinary graph*, i.e., a graph where every

[1] That is, find the minimum number of sets that cover the entire universe.

[2] Note that d and r are dual parameters in the sense that if the input is $\{S_1, \ldots, S_m\}$ and we define $T_i = \{j : i \in S_j\}$ then $d = \max_j |S_j|$ and $r = \max_i |T_i|$.

edge just has two endpoints. With this interpretation, the graph is being presented in the arbitrary order model.

2. *Second Interpretation:* A sequence of adjacency lists (where the adjacency list for a given node includes all the hyperedges that include that node) on a graph with m nodes of maximum degree d and n hyperedges of maximum size r. In this interpretation, if every element appears in exactly $r = 2$ sets, then this corresponds to an ordinary graph where each element corresponds to an edge and each set corresponds to a node. With this interpretation, the graph is being presented in the adjacency list model.

Under the first interpretation, the `Max Coverage` problem and the `Max Unique Coverage` problem when all sets have size exactly 2 naturally generalize the problem of finding a maximum matching in an ordinary graph in the sense that if there exists a matching with at least k edges, the optimum solution to either `Max Coverage` and `Max Unique Coverage` will be a matching. There is a large body of work on graph matchings in the data stream model [3, 12, 23, 24, 28, 31, 34, 35, 43, 44, 49–51, 55, 66] including work specifically on solving the problem exactly if the matching size is bounded [18, 20]. More precisely, `Max Coverage` corresponds to the partial vertex cover problem [54]: what is the maximum number of edges that can be covered by selecting k nodes. For larger sets, the `Max Coverage` and `Max Unique Coverage` are at least as hard as finding partial vertex covers and matching in hypergraphs.

Under the second interpretation, when all elements have multiplicity 2, then the problem `Max Unique Coverage` corresponds to finding the capacitated maximum cut, i.e., a set of at most k vertices such that the number of edges with exactly one endpoint in this set is maximized. In the offline setting, Ageev and Sviridenko [1] and Gaur et al. [33] presented a 2 approximation for this problem using linear programming and local search respectively. The (uncapacitated) maximum cut problem was been studied in the data stream model by Kapralov et al. [45–47]; a 2-approximation is trivial in logarithmic space[3] but improving on this requires space that is polynomial in the size of the graph. The capacitated problem is a special case of the problem of maximizing a non-monotone sub-modular function subject to a cardinality constraint. This general problem has been considered in the data stream model [8, 13, 16, 37] but in that line of work it is assumed that there is oracle access to the function being optimized, e.g., given any set of nodes, the oracle will return the number of edges cut. Alaluf et al. [4] presented a $2+\epsilon$ approximation in this setting, assuming exponential post-processing time. In contrast, our algorithm does not assume an oracle while obtaining a $1+\epsilon$ approximation (and also works for the more general problem `Max Unique Coverage`).

1.1 Our Results

Our main results are the following single-pass streaming algorithms[4]:

(A) Bounded Set Cardinality. If all sets have size at most d, there exists a $\tilde{O}(d^{d+1}k^d)$ space data stream algorithm that solves `Max Unique Coverage` and `Max Coverage` exactly. We show that this is nearly optimal in the sense that any exact algorithm requires $\Omega(k^d)$ space for constant d.

(B) Bounded Multiplicity. If every element appears in at most r sets, we present the following algorithms:

[3] It suffices to count the number of edges M since there is always a cut whose size is at least $M/2$.
[4] Throughout we use $\tilde{O}$ to denote that logarithmic factors of m and n are being omitted.

- (B1) `Max Unique Coverage`: There exists a $1+\epsilon$ approximation using $\tilde{O}(\epsilon^{-4}k^3r)$ space.
- (B2) `Max Coverage`: There exists a $1+\epsilon$ approximation algorithm using $\tilde{O}(\epsilon^{-3}k^2r)$ space.

In contrast to the above results, when d and r are arbitrary, any constant pass $1+\epsilon$ approximation algorithm for either problem requires $\Omega(\epsilon^{-2}m)$ space [6].[5] We also generalize of lower bound for `Max Coverage` [61] to `Max Unique Coverage` to show that any constant-pass algorithm with an approximation better than $e^{1-1/k}$ requires $\Omega(m/k^2)$ space. We also present a single-pass algorithm with an $O(\log \min(k, r))$ approximation for `Max Unique Coverage` using $\tilde{O}(k^2)$ space, i.e., the space is independent of r and d but the approximation factor depends on r. This algorithm is a simple combination of a `Max Coverage` algorithm due to McGregor and Vu [61] and an algorithm for `Max Unique Coverage` in the offline setting due to Demaine et al. [25]. Finally, our `Max Coverage` result (B2) algorithm also yields a new multi-pass result for a parameterized version of the streaming `Set Cover` problem. We will also show that results (A) and (B2) can also be made to handle stream deletions. The generalization for result (A) that we present requires space that scales with k^{2d} rather than k^d. However, in subsequent work we have shown that space the scales with k^d is also sufficient in the insert/delete setting.

1.2 Technical Summary and Comparisons

Technical Summary. Our results are essentially streamable kernelization results, i.e., the algorithm "prunes" the input (in the case of `Max Unique Coverage` and `Max Coverage` this corresponds to ignoring some of the input sets) to produce a "kernel" in such a way that a) solving the problem optimally on the kernel yields a solution that is as good (or almost as good) as the optimal solution on the original input and b) the kernel can be constructed in the data stream model and is sufficiently smaller than the original input such that it is possible to find an optimal solution for the kernel in significantly less time than it would take to solve on the original input. In the field of fixed parameter tractability, the main requirement is that the kernel can be produced in polynomial time. In the growing body of work on streaming kernelization [17–19] the main requirement is that the kernel can be constructed using small space in the data stream model. Our results fits in with this line of work and the analysis requires numerous combinatorial insights into the structure of the optimum solution for `Max Unique Coverage` and `Max Coverage`.

Our technical contributions can be outlined as follows.

- Result (A) relies on a key combinatorial lemma. This lemma provides a rule to discard sets such that there is an optimum solution that does not contain any of the discarded sets. Furthermore, the number of stored sets can be bounded in terms of k and d.
- Result (B1) uses the observation that each set of any optimal solution intersects some maximal collection of disjoint sets. The main technical step is to demonstrate that storing a small number of intersecting sets, in terms of k and r, suffices to preserve the optimal solution.
- Result (B2) is based on a very simple idea of first collecting the largest $O(rk/\epsilon)$ sets and then solving the problem optimally on these sets. This can be done in a space efficient manner using existing sketch for F_0 estimation in the case of `Max Coverage`. While the

[5] The lower bound result by Assadi [6] was for the case of `Max Coverage` but we will explain that it also applies in the case of `Max Unique Coverage`.

approach is simple, showing that it yields the required approximations requires some work that builds on a recent result by Manurangsi [54]. We also extend the algorithm to the model where sets can be inserted and deleted.

Comparison to Related Work. In the context of streaming algorithms, for the `Max Coverage` problem, McGregor and Vu [60] showed that any approximation better than $e/(e-1)$ requires $\Omega(m/k^2)$ space. For the more general problem of streaming submodular maximization subject to a cardinality constraint, Feldman et al. [32] very recently showed a stronger lower bound that any approximation better than 2 requires $\Omega(m)$ space. Our results provide a route to circumvent these bounds via parameterization on k, r, and d.

Result (B2) also leads to a parameterized algorithm for streaming `Set Cover`. This new algorithm uses $\tilde{O}(rk^2n^\delta + n)$ space which improves upon the algorithm by Har-Peled et al. [36] that uses $\tilde{O}(mn^\delta + n)$ space, where k is an upper bound for the size of the minimum set cover, in the case $rk^2 \ll m$. Both algorithms use $O(1/\delta)$ passes and yield an $O(1/\delta)$ approximation.

In the context of offline parameterized algorithms, Bonnet et al. [10] showed that `Max Coverage` is fixed-parameter tractable in terms of k and d. However, their branching-search algorithm cannot be implemented in the streaming setting. Misra et al. [62] showed that the maximum unique coverage problem in which the aim is to maximize the number of uniquely covered elements u (without any restriction on the number of sets) admits a kernel of size 4^u. On the other hand, they showed that the budgeted version of this problem (where each element has a profit and each set has a cost and the goal is maximize the profit subject to a budget constraint) is $W[1]$-hard when parameterized by the budget[6]. In this context, our result shows that a parameterization on both the maximum set size d and the budget k is possible (at least when all costs and profits are unit).

2 Preliminaries

2.1 Notation and Parameters

Throughout the paper, m will denote the number of sets, n will denote the size of the universe, and k will denote the maximum number of sets that can be used in the solution. Given input sets $S_1, S_2, \ldots, S_m \subseteq [n]$, let

$$d = \max_i |S_i|$$

be the maximum set size and let

$$r = \max_j |\{i : j \in S_i\}|$$

be the maximum number of sets that contain the same element.

Suppose C is a collection of sets. We let $F(C)$ (and $G(C)$) be the set of elements covered (and uniquely covered) by an optimal solution in C. Furthermore, let $f(C) = |F(C)|$ and $g(C) = |G(C)|$. In other words, $f(C)$ is the maximum number of elements that can be covered by k sets. Similarly, $g(C)$ is the maximum number of elements that can be uniquely covered by k sets. Furthermore, let $\psi(C)$ and $\tilde{\psi}(C)$ be the set of elements covered and uniquely covered respectively by the sets in C.

[6] In the `Max Unique Coverage` problem that we consider, all costs and profits are one and the budget is k.

To ease the notation, if C is a collection of set and S is a set, we often use $C - S$ to denote $C \setminus \{S\}$ and $C + S$ to denote $C \cup \{S\}$.

We use M to denote the collection of all sets in the stream. Therefore, the optimal value to `Max Coverage` and `Max Unique Coverage` are $f(M)$ and $g(M)$ respectively.

Throughout this paper, we say an algorithm is correct with high probability if the probability of failure is inversely polynomial in m.

2.2 Sketches and Subsampling

Coverage Sketch. Given a vector $x \in \mathbb{R}^n$, $F_0(x)$ is defined as the number of elements of x which are non-zero. If given a subset $S \subset \{1, \ldots, n\}$, we define $x_S \in \{0,1\}^n$ to be the characteristic vector of S (i.e., $x_i = 1$ iff $i \in S$) then given sets $S_1, S_2, \ldots$ note that $F_0(x_{S_1} + x_{S_2} + \ldots)$ is exactly the number of elements covered by $S_1 \cup S_2 \cup \ldots$. We will use the following result for estimating F_0.

▶ **Theorem 1** (F_0 Sketch [9, 21]). *Given a set $S \subseteq [n]$, there exists an $\tilde{O}(\epsilon^{-2} \log \delta^{-1})$-space algorithm that constructs a data structure $\mathcal{M}(S)$ (called an F_0 sketch of S). The sketch has the property that the number of distinct elements in a collection of sets $S_1, S_2, \ldots, S_t$ can be approximated up to a $1+\epsilon$ factor with probability at least $1-\delta$ provided the collection of F_0 sketches $\mathcal{M}(S_1), \mathcal{M}(S_2), \ldots, \mathcal{M}(S_t)$.*

Note that if we set $\delta \ll 1/(\text{poly}(m) \cdot \binom{t}{k})$ in the above result we can try each collection of k sets amongst $S_1, S_2, \ldots, S_t$ and get a $1+\epsilon$ approximation for the coverage of each collection with high probability.

Unique Coverage Sketch. For unique coverage, our sketch of a set corresponds to subsampling the universe via some hash function $h : [n] \to \{0,1\}$ where h is chosen randomly such that for each i, $\Pr[h(i) = 1] = p$ for some appropriate value p. Specifically, rather processing an input set S, we process $S' = \{i \in S : h(i) = 1\}$. Note that $|S'|$ has size $p|S|$ in expectation. This approach was use by McGregor and Vu [61] in the context of `Max Coverage` and it extends easily to `Max Unique Coverage`; see Section 7. The consequence is that if there is a streaming algorithm that finds a t approximation, we can turn that algorithm into a $t(1+\epsilon)$ approximation algorithm in which we can assume that $\text{OPT} = O(\epsilon^{-2} k \log m)$ with high probability by running the algorithm on a subsampled sets rather than the original sets. Note that this also allows us to assume input sets have size $O(\epsilon^{-2} k \log m)$ since $|S'| \leq \text{OPT}$. Hence each "sketches" set can be stored using $B = O(\epsilon^{-2} k \log m \log n)$ bits.

An Algorithm with $\tilde{O}(\epsilon^{-2}mk)$ Memory. We will use the above sketches in a more interesting context later in the paper, but note that they immediately imply a trivial algorithmic result. Consider the naive algorithm that stores every set and finds the best solution; note that this requires exponential time. We note that since we can assume $\text{OPT} = O(\epsilon^{-2} k \log m)$, each set has size at most $O(\epsilon^{-2} k \log m)$. Hence, we need $\tilde{O}(\epsilon^{-2}mk)$ memory to store all the sets. This approach was noted in [61] in the context of `Max Coverage` but also apples to `Max Unique Coverage`. We will later show that for a $1+\epsilon$ approximation, the above trivial algorithm is optimal up to polylogarithmic factors for constant k.

3 An Exact Algorithm

Algorithm. Our algorithm, though perhaps non-intuitive, is simple to state:

1. Initialize X to be an empty collection of sets. Let $b = d(k-1)$.
2. Let X_a be the sub-collection of X that contains sets of size a.
3. For each set S in the stream: Suppose $|S| = a$. Add S to X if there does not exist $T \subseteq S$ that occurs as a subset of $(b+1)^{d-|T|}$ sets of X_a.
4. Post-processing: Return the best solution C in X.

Analysis. Our algorithm relies on the following combinatorial lemma.

▶ **Lemma 2.** *Let $W = \{S_1, S_2, \ldots\}$ be a collection of distinct sets where each $S_i \subseteq [n]$ and $|S_i| = a$. Suppose for all $T \subseteq \psi(W)$ with $|T| \leq a$ there exist at most*

$$\ell_{|T|} := (b+1)^{a-|T|}$$

sets in W that contain T. Furthermore, suppose there exists a set T^* such that this inequality is tight. Then, for all $B \subseteq \psi(W)$ disjoint from T^* with $|B| \leq b$ there exists a set $Y \in W$ such that $T^* \subseteq Y$ and $|Y \cap B| = 0$.

Proof. If $|T^*| = a$ then $T^* \in W$, then we can simply set $Y = T^*$. Henceforth, assume $|T^*| < a$. Consider the $\ell_{|T^*|}$ sets in W that are supersets of T^*. Call this collection W'. For any $x \in B$, there are at most $\ell_{|T^*|+1}$ sets that include $T^* \cup \{x\}$. Since there are b choices for x, at most

$$b\ell_{|T^*|+1} = b(b+1)^{a-|T^*|-1} < (b+1)^{a-|T^*|} = \ell_{|T^*|}$$

sets in W' contain an element in B. Hence, at least one set Y in W' does not contain any element in B. ◀

We show that the algorithm indeed obtains an exact kernel for the problems. Recall that M is the collection of all sets in the stream, i.e., the optimal solution has size $f(M)$.

▶ **Theorem 3.** *The output of the algorithm is optimal. In particular, $f(C) = f(M)$ and $g(C) = g(M)$.*

Proof. Recall that X is the collection of all stored sets. We define

$$C_i = M \setminus \{\text{the first } i \text{ sets in the stream that are not stored in } X\}.$$

Clearly, $f(C_0) = f(M)$. Now, suppose there exists $i \geq 1$ such that $f(C_i) < f(M)$. Let i be the smallest such index. Let $\mathcal{O}$ be an optimal solution of C_{i-1} (note that $\mathcal{O}$ is also an overall optimal solution based on the minimal assumption on i). Let S be the ith set that was not stored in X. If $S \notin \mathcal{O}$ then we have a contradiction since $f(C_i) = f(C_{i-1}) = f(M)$. Thus, assume $S \in \mathcal{O}$. Suppose $|S| = a$.

▷ Claim 4. There exists Y in X_a such that $f(\mathcal{O} - S + Y) \geq f(\mathcal{O})$.

Proof. Note that S was not stored because there existed $T^* \subseteq S$ such that T^* was a subset of $(b+1)^{d-|T^*|}$ sets in X_a. Consider the set $B = \psi(\mathcal{O}) \setminus S$. Clearly, $B \cap T^* = \emptyset$ and $|B| \leq d(k-1)$. By Lemma 2, there is a set Y in X_a such that $Y \cap B = \emptyset$.

Let $Y' = Y \setminus S$ and $S' = S \setminus Y$. Note that $|Y'| = |S'|$ since $|Y| = |S|$. Define indicator variables $\alpha_z = 1$ iff $z \in \psi(\mathcal{O} - S + Y)$ and $\beta_z = 1$ iff $z \in \psi(\mathcal{O})$. Note that

$$\begin{aligned}(z \in Y \cap S \text{ or } z \notin Y \cup S) &\implies (\alpha_z = \beta_z)\,,\\ (z \in Y') &\implies (\alpha_z = 1)\,,\\ (z \in Y') &\implies (\beta_z = 0)\,,\end{aligned}$$

where the last equation uses the fact that Y' is disjoint from $\psi(\mathcal{O})$. Then

$$\begin{aligned}|\psi(\mathcal{O}-S+Y)| &= \sum_{z\in Y'}\alpha_z + \sum_{z\in Y\cap S}\alpha_z + \sum_{z\in S'}\alpha_z + \sum_{z\notin Y\cup S}\alpha_z \\ &\geq \left(|Y'| + \sum_{z\in Y'}\beta_z\right) + \sum_{z\in Y\cap S}\beta_z + \left(-|S'| + \sum_{z\in S'}\beta_z\right) + \sum_{z\notin Y\cup S}\beta_z \\ &= \sum_{z\in Y'}\beta_z + \sum_{z\in Y\cap S}\beta_z + \sum_{z\in S'}\beta_z + \sum_{z\notin Y\cup S}\beta_z = |\psi(\mathcal{O})|\,. \end{aligned}$$

◀

Thus, $f(C_i) \geq f(\mathcal{O}) = f(M)$ which is a contradiction. Hence, there is no such i and the claim follows. The proof for unique coverage is almost identical: for the analogous claim we define indicator variables $\tilde{\alpha}_z = 1$ iff $z \in \tilde{\psi}(\mathcal{O} - S + Y)$ and $\tilde{\beta}_z = 1$ iff $z \in \tilde{\psi}(\mathcal{O})$. The proof goes through with α and β replaced by $\tilde{\alpha}$ and $\tilde{\beta}$ since it is still the case that

$$\begin{aligned}(z \in Y \cap S \text{ or } z \notin Y \cup S) &\implies \left(\tilde{\alpha}_z = \tilde{\beta}_z\right), \\ (z \in Y') &\implies (\tilde{\alpha}_z = 1), \\ (z \in Y') &\implies \left(\tilde{\beta}_z = 0\right), \end{aligned}$$

where now the last two equations use the fact that Y' is disjoint from $\psi(\mathcal{O})$. ◀

▶ **Lemma 5.** *The space used by the algorithm is $\tilde{O}(d^{d+1}k^d)$.*

Proof. Recall that one of the requirements for a set S to be added to X is that the number of sets in $X_{|S|}$ that are supersets of any subset of S of size t is at most $(b+1)^{d-t}$. This includes the empty subset and since every set in $X_{|S|}$ is a superset of the empty set, we deduce that $|X_{|S|}| \leq (b+1)^d = O((dk)^d)$. Since each set needs $\tilde{O}(d)$ bits to store, and $|X| = \sum_{a=1}^{d} |X_a| \leq O(d^d k^d)$, the total space is $\tilde{O}(d^{d+1}k^d)$. ◀

We summarize the above as a theorem.

▶ **Theorem 6.** *There exist deterministic single-pass algorithms using $\tilde{O}(k^d d^{d+1})$ space that yields an exact solution to* `Max Coverage` *and* `Max Unique Coverage`*.*

Handling Insertion-Deletion Streams. We outline another exact algorithm that works for insertion-deletion streams, however with a worse space bound $\tilde{O}((kd)^{2d})$, in Section 6.1.

▶ **Theorem 7.** *There exist randomized single-pass algorithms using $\tilde{O}(d^{2d}k^d)$ space and allowing deletions that w.h.p. yield an exact solution to* `Max Coverage` *and* `Max Unique Coverage`*.*

[7]

4 Approximation Algorithms

In this section, we present a variety of different approximation algorithms where the space used by the algorithm is independent of d but, in some cases, may depend on r. The first algorithm uses $\tilde{O}(\epsilon^{-4}k^3 r)$ memory and obtains a $1+\epsilon$ approximation to both problems. The second algorithm uses $\tilde{O}(\epsilon^{-3}k^2 r)$ memory and obtains a $1+\epsilon$ approximation to `Max Coverage` and a $2+\epsilon$ approximation to `Max Unique Coverage`; it can also be extended to streams with deletions.

[7] This improves upon our earlier result in the ICDT version of the paper that uses $\tilde{O}(d^{2d}k^d)$ space.

4.1 A $1+\epsilon$ Approximation

Given a collection of sets $C = \{S_1, S_2, \ldots, S_m\}$, we say a sub-collection $C' \subset C$ is a *matching* if the sets in C' are mutually disjoint. C' is a maximal matching if there does not exist $S \in C \setminus C'$ such that S is disjoint from all sets in C'.

▶ **Lemma 8.** *For any input C, let $O \subset C$ be an optimal solution for either the* `Max Coverage` *or* `Max Unique Coverage` *problem. Let M_i be a maximal matching amongst the input set of size i. Then every set of size i in O intersects with some set in M_i.*

Proof. Let $S \in O$ have size i. If it was disjoint from all sets in M_i then it could be added to M_i and the resulting collection would still be a matching. This violates the assumption that M_i is maximal. ◀

The next lemma extends the above result to show that we can potentially remove many sets from each M_i and still argue that there is an optimal solution for the original instance amongst the sets that intersect a set in some M_i.

▶ **Lemma 9.** *Consider an input of sets of size at most d. For $i \in [d]$, let M_i be a maximal matching amongst the input set of size i and let M'_i be an arbitrary subset of M_i of size* $\min(k+dk, |M_i|)$. *Let D_i be the collection of all sets that intersect a set in M'_i. Then $\bigcup_i (D_i \cup M'_i)$ contains an optimal solution to both the* `Max Unique Coverage` *and* `Max Coverage` *problem.*

Proof. If $|M_i| = |M'_i|$ for all $1 \le i \le d$ then the result follows from Lemma 8. If not, let $j = \max\{i \in [d] : |M_i| > |M'_i|\}$. Let $\mathcal{O}$ be an optimal solution and let $\mathcal{O}_i$ be all the sets in $\mathcal{O}$ of size i. We know that every set in $\mathcal{O}_d \cup \mathcal{O}_{d-1} \cup \ldots \cup \mathcal{O}_{j+1}$ is in

$$\bigcup_{i \ge j+1} (D_i \cup M'_i) = \bigcup_{i \ge j+1} (D_i \cup M_i) \ .$$

Hence, the number of elements (uniquely) covered by $\mathcal{O}$ is at most the number of elements (uniquely) covered by $\mathcal{O}_d \cup \mathcal{O}_{d-1} \cup \ldots \cup \mathcal{O}_{j+1}$ plus kj since every set in $\mathcal{O}_j \cup \ldots \cup \mathcal{O}_1$ (uniquely) covers at most j additional elements. But we can (uniquely) cover at least the number of elements (uniquely) covered by $\mathcal{O}_d \cup \mathcal{O}_{d-1} \cup \ldots \cup \mathcal{O}_{j+1}$ plus kj. This is because M_j contains $k + dk$ disjoint sets of size j and at least $k + dk - kd = k$ of these are disjoint from all sets in $\mathcal{O}_d \cup \mathcal{O}_{d-1} \cup \ldots \cup \mathcal{O}_{j+1}$. Hence, there is a solution amongst $\bigcup_{i \ge j} (D_i \cup M'_i)$ that is at least as good as $\mathcal{O}$ and hence is also optimal. ◀

The above lemma suggests an exact algorithm that stores the sets in $\bigcup_i (D_i \cup M'_i)$ and find the optimum solution among these sets. In particular, we construct matchings of each size greedily up to the appropriate size and store all intersecting sets. Note that since each element belongs to at most r sets, the total space is $\tilde{O}(d^2kr)$. Applying the sub-sampling framework, we have $d \le \mathrm{OPT} = O(k/\epsilon^2 \log m)$ and the approximation factor becomes $1+\epsilon$.

▶ **Theorem 10.** *There exists a randomized one-pass algorithm using $\tilde{O}(\epsilon^{-4}k^3r)$ space that finds a $1+\epsilon$ approximation to* `Max Unique Coverage` *and* `Max Coverage`.

4.2 A More Efficient $1+\epsilon$ Approximation for Maximum Coverage

In this section, we generalize the approach of Manurangsi [54] and combine that with the F_0-sketching technique to obtain a $1+\epsilon$ approximation using $\tilde{O}(\epsilon^{-3}k^2r)$ space for maximum coverage. This saves a factor k/ϵ and the generalized analysis might be of independent interest. Let $\mathrm{OPT} = \psi(\mathcal{O})$ denote the optimal coverage of the input stream.

Manurangsi [54] showed that for the maximum k-vertex cover problem, the $\Theta(k/\epsilon)$ vertices with highest degrees form a $1+\epsilon$ approximation kernel for the maximum k vertex coverage problem. That is, there exist k vertices among those that cover $(1-\epsilon)\,\mathrm{OPT}$ edges. We now consider a set system in which an element belongs to at most r sets (this can also be viewed as a hypergraph where each set corresponds to a vertex and each element corresponds to a hyperedge; we then want to find k vertices that touch as many hyperedges as possible).

We begin with the following lemma that generalizes the aforementioned result in [54]. We may assume that $m > rk/\epsilon$ since otherwise, we can store all the sets.

▶ **Lemma 11.** *Suppose $m > \lceil rk/\epsilon \rceil$. Let K be the collection of $\lceil rk/\epsilon \rceil$ sets with largest sizes (tie-broken arbitrarily). There exist k sets in K that cover $(1-\epsilon)\,\mathrm{OPT}$ elements.*

Proof. Let $\mathcal{O}$ denote the collection of k sets in some optimal solution. Let $\mathcal{O}^{in} = \mathcal{O} \cap K$ and $\mathcal{O}^{out} = \mathcal{O} \setminus K$. We consider a random subset $Z \subset K$ of size $|\mathcal{O}^{out}|$. We will show that the sets in $Z \cup \mathcal{O}^{in}$ cover $(1-\epsilon)\,\mathrm{OPT}$ elements in expectation; this implies the claim.

Let $[\mathcal{E}]$ denote the indicator variable for event $\mathcal{E}$. We rewrite

$$|\psi(Z \cup \mathcal{O}^{in})| = |\psi(\mathcal{O}^{in})| + |\psi(Z)| - |\psi(\mathcal{O}^{in}) \cap \psi(Z)| \ .$$

Furthermore, the probability that we pick a set S in K to add to Z is

$$p := \frac{|\mathcal{O}^{out}|}{|K|} \leq \frac{k}{kr/\epsilon} = \frac{\epsilon}{r} \ .$$

Next, we upper bound $\mathrm{E}\left[|\psi(\mathcal{O}^{in}) \cap \psi(Z)|\right]$. We have

$$\mathrm{E}\left[|\psi(\mathcal{O}^{in}) \cap \psi(Z)|\right] \leq \sum_{u \in \psi(\mathcal{O}^{in})} \sum_{S \in K : u \in S} \Pr\left[S \in Z\right] \leq \sum_{u \in \psi(\mathcal{O}^{in})} rp \leq |\psi(\mathcal{O}^{in})| \cdot \epsilon \ .$$

We lower bound $\mathrm{E}\left[|\psi(Z)|\right]$ as follows.

$$\begin{aligned}
\mathrm{E}\left[|\psi(Z)|\right] &\geq \mathrm{E}\left[\sum_{S \in K} \left(|S|[S \in Z] - \sum_{S' \in K \setminus \{S\}} |S \cap S'|[S \in Z \wedge S' \in Z] \right)\right] \\
&\geq \sum_{S \in K} \left(|S|p - \sum_{S' \in K \setminus \{S\}} |S \cap S'|p^2 \right) \\
&\geq \sum_{S \in K} \left(|S|p - (r-1)|S|p^2 \right) \geq p(1-pr) \sum_{S \in K} |S| \geq p(1-\epsilon) \sum_{S \in K} |S| \ . \qquad (1)
\end{aligned}$$

In the above derivation, the second inequality follows from the observation that

$$\Pr\left[S \in Z \wedge S' \in Z\right] \leq p^2 \ .$$

The third inequality is because $\sum_{S' \in K \setminus \{S\}} |S \cap S'| \leq (r-1)|S|$ since each element belongs to at most r sets.

For all $S \in K$, we must have

$$|S| \geq \frac{\sum_{Y \in \mathcal{O}^{out}} |Y|}{|\mathcal{O}^{out}|} \geq \frac{|\psi(\mathcal{O}^{out})|}{|\mathcal{O}^{out}|} \ .$$

Thus,

$$\mathrm{E}\left[|\psi(Z)|\right] \geq p\,(1-\epsilon)\,|K| \frac{|\psi(\mathcal{O}^{out})|}{|\mathcal{O}^{out}|} = p\,(1-\epsilon)\,\frac{|\psi(\mathcal{O}^{out})|}{p} = (1-\epsilon)|\psi(\mathcal{O}^{out})| \ .$$

Putting it together,

$$\mathrm{E}\left[|\psi(Z \cup \mathcal{O}^{in})|\right] \geq |\psi(\mathcal{O}^{in})| + (1-\epsilon)|\psi(\mathcal{O}^{out})| - |\psi(\mathcal{O}^{in})| \cdot \epsilon \geq (1-\epsilon)\,\mathrm{OPT} \ . \qquad \blacktriangleleft$$

With the above lemma in mind, the following algorithm's correctness is immediate.

1. Store F_0-sketches of the $\lceil kr/\epsilon \rceil$ largest sets, where the failure probability of the sketches is set to $\frac{1}{\text{poly}(n)\binom{m}{k}}$.
2. At the end of the stream, return the k sets with the largest coverage based on the estimates given by the F_0-sketches.

We restate our result as a theorem.

▶ **Theorem 12.** *There exists a randomized one-pass, $\tilde{O}(k^2r/\epsilon^3)$-space, algorithm that with high probability finds a $1+\epsilon$ approximation to* `Max Coverage`.

Obtaining a $2+\epsilon$ approximation to `Max Unique Coverage`. We note that finding the best solution to `Max Unique Coverage` in K will yield a $2+\epsilon$ approximation. This is a worse approximation than that of the previous subsection. However, we save a factor of k/ϵ in memory. Furthermore, this approach also allows us to handle streams with deletions.

To see that we get a $2+\epsilon$ approximation to `Max Unique Coverage`. Note that $g(Z \cup \mathcal{O}^{in}) \geq \frac{1}{2}\left(g(\mathcal{O}^{in}) + g(Z)\right)$. Furthermore, a similar derivation shows $\mathrm{E}\left[|\tilde{\psi}(Z)|\right] \geq (1-\epsilon)|\tilde{\psi}(\mathcal{O}^{out})|$. Specifically, in the derivation in Eq. 1, we can simply replace ψ with $\tilde{\psi}$. This gives us $g(K) \geq (1/2-\epsilon)g(\mathcal{O})$.

Extension to Insert/Delete Streams. The result can be extended to the case where sets are inserted and deleted. For the full details, see Section 6.2.

4.3 An $O(\log \min(k, r))$ Approximation for Unique Coverage

We now present an algorithm whose space does not depend on r but the result comes at the cost of increasing the approximation factor to $O(\log(\min(k, r)))$. It also has the feature that the running time is polynomial in k in addition to being polynomial in m and n.

The basic idea is as follows: We consider an existing algorithm that first finds a 2.01 approximation C to `Max Coverage`. It then finds the best solution of `Max Unique Coverage` among the sets in C.

▶ **Theorem 13.** *There exists a randomized one-pass, $\tilde{O}(k^2)$-space, algorithm that with high probability finds a $O(\log \min(k, r))$ approximation to* `Max Unique Coverage`.

Proof. From previous work [8,61], we can find a 2.01 approximation C to `Max Coverage` using $\tilde{O}(k)$ memory. Note that their algorithm maintains a collection C of k sets during the stream. Demaine et al. [25] proved that that if Q is the best solution to `Max Unique Coverage` among the sets in C, then Q is an $O(\log \min(k, r))$ approximation to `Max Unique Coverage`. In fact, they presented a polynomial time algorithm to find Q from C such that the number of uniquely covered elements is at least

$$\Omega(1/\log k) \cdot |\psi(C)| \geq \Omega(1/\log k) \cdot 1/2.01 \cdot f(M) \geq \Omega(1/\log k) \cdot g(M) \ .$$

Note that storing each set in C requires $\tilde{O}(d)$ memory. Hence, the total memory is $\tilde{O}(kd)$. Applying the sub-sampling framework, we obtain an $\tilde{O}(k^2)$ memory algorithm.

◀

4.4 Application to Parameterized Set Cover

We parameterize the set cover problem as follows. Given a set system, either A) output a set cover of size αk if OPT $\leq k$ where α the approximation factor or B) correctly declare that a set cover of size k does not exist.

▶ **Theorem 14.** *For $0 < \delta < 1$, there exists a randomized, $O(1/\delta)$-pass, $\tilde{O}(rk^2n^\delta + n)$-space, algorithm that with high probability finds a $O(1/\delta)$ approximation to the parameterized* `Set Cover` *problem.*

Proof. In each pass, we run the algorithm in Theorem 12 with parameters k and $\epsilon = 1/n^{\delta/3}$ on the remaining uncovered elements. The space use is $\tilde{O}(rk^2n^\delta + n)$. Here, we need additional $\tilde{O}(n)$ space to keep track of the remaining uncovered elements.

Note that if OPT $\leq k$, after each pass, the number of uncovered elements is reduced by a factor $1/n^{\delta/3}$. This is because if n' is the number of uncovered elements at the beginning of a pass, then after that pass, we cover all but at most $n'/n^{\delta/3}$ of those elements. After i passes, the number of remaining uncovered elements is $O(n^{1-i\delta/3})$; we therefore use at most $O(1/\delta)$ passes until we are done. At the end, we have a set cover of size $O(k/\delta)$.

If after $\omega(1/\delta)$ passes, there are still remaining uncovered elements, we declare that such a solution does not exist. ◀

Our algorithm improves upon the algorithm by Har-Peled et al. [36] that uses $\tilde{O}(mn^\delta + n)$ space for when $rk^2 \ll m$. Both algorithms yield an $O(1/\delta)$ approximation and use $O(1/\delta)$ passes.

5 Lower Bounds

5.1 Lower Bounds for Exact Solutions

As observed earlier, any exact algorithm for either the `Max Coverage` or `Max Unique Coverage` problem on an input where all sets have size d will return a matching of size k if one exists. However, by a lower bound due to Chitnis et al. [18] we know that determining if there exists a matching of size k in a single pass requires $\Omega(k^d)$ space. This immediately implies the following theorem.

▶ **Theorem 15.** *Any single-pass algorithm that solves* `Max Coverage` *or* `Max Unique Coverage` *exactly with probability at least $9/10$ requires $\Omega(k^d)$ space.*

5.2 Lower bound for a $e^{1-1/k}$ approximation

The strategy is similar to previous work on `Max Coverage` [60, 61]. However, we need to argue that the relevant probabilistic construction works for all collections of fewer than k sets since the unique coverage function is not monotone.

We make a reduction from the communication problem k-player set disjointness, denoted by `DISJ`(m, k). In this problem, there are k players where the ith player has a set $S_i \subseteq [m]$. It is promised that exactly one of the following two cases happens a) NO instance: All the sets are pairwise disjoint and b) YES instance: There is a unique element $v \in [m]$ such that $v \in S_i$ for all $i \in [k]$ and all other elements belong to at most one set. The (randomized) communication complexity (in the one-way model or the blackboard model), for some large enough constant success probability, of the above problem is $\Omega(m/k)$ even if the players may use public randomness [14]. We can assume that $|S_1 \cup S_2 \cup \ldots \cup S_k| \geq m/4$ via a padding argument.

▶ **Theorem 16.** *Any constant-pass randomized algorithm with an approximation better than* $e^{1-1/k}$ *to* `Max Unique Coverage` *requires* $\Omega(m/k^2)$ *space.*

Proof. For each $i \in [m]$, let $\mathcal{P}_i$ be a random partition of $[n]$ into k sets $V_1^i, \ldots, V_k^i$ such that an element in the universe $U = [n]$ belongs to exactly one of these sets uniformly at random. In particular, for all $i \in [m]$ and $v \in U$,

$$\Pr\left[v \in V_j^i \wedge (\forall j' \neq j, v \notin V_{j'}^i)\right] = 1/k \ .$$

The partitions are chosen independently using public randomness before receiving the input. For each player j, if $i \in S_j$, then they put V_j^i in the stream. Note that the stream consists of $\Theta(m)$ sets.

If the input is a NO instance, then for each $i \in [m]$, there is at most one set V_j^i in the stream. Therefore, for each element $v \in [n]$ and any collection of $\ell \leq k$ sets $V_{j_1}^{i_1}, \ldots, V_{j_\ell}^{i_\ell}$ in the stream,

$$\Pr\left[v \text{ is uniquely covered by } V_{j_1}^{i_1}, \ldots, V_{j_\ell}^{i_\ell}\right] = \ell/k \cdot (1-1/k)^{\ell-1} \leq \ell/k \cdot e^{-(\ell-1)/k} \ .$$

Therefore, in expectation, $\mu_\ell := \mathrm{E}\left[g(\{V_{j_1}^{i_1}, \ldots, V_{j_\ell}^{i_\ell}\})\right] \leq \ell/k \cdot e^{-(\ell-1)/k} n$. By an application of Hoeffding's inequality,

$$\begin{aligned}\Pr\left[g(\{V_{j_1}^{i_1} \cup \ldots \cup V_{j_\ell}^{i_\ell}\}) > \mu_\ell + \epsilon e^{-(k-1)/k} \cdot n\right] &\leq \exp\left(-2\epsilon^2 e^{-2(\ell-1)/k} n\right) \\ &\leq \exp\left(-\Omega(\epsilon^2 n)\right) \leq \frac{1}{m^{10k}} \ .\end{aligned}$$

The last inequality follows by letting $n = \Omega(\epsilon^{-2} k \log m)$. The following claim shows that for large k, in expectation, picking k sets is optimal in terms of unique coverage.

▶ **Lemma 17.** *The function* $g(\ell) = \ell/k \cdot e^{-(\ell-1)/k} n$ *is increasing in the interval* $(-\infty, k]$ *and decreasing in the interval* $[k, +\infty)$*.*

Proof. We take the partial derivative of g with respect to ℓ

$$\frac{\partial g}{\partial \ell} = \frac{e^{(1-\ell)/k}(k-\ell)}{k^2} \cdot n$$

and observe that it is non-negative if and only if $\ell \leq k$. ◀

By appealing to the union bound over all $\binom{m}{1} + \ldots + \binom{m}{k-1} + \binom{m}{k} \leq O(m^{k+1})$ possible collections $\ell \leq k$ sets, we deduce that with high probability, for all collections of $\ell \leq k$ sets $S_1, \ldots, S_\ell$,

$$\begin{aligned}g(\{S_1, \ldots, S_\ell\}) \leq \mu_\ell + \epsilon e^{-(k-1)/k} \cdot n &\leq \ell/k \cdot e^{-(\ell-1)/k} n + \epsilon e^{-(k-1)/k} \cdot n \\ &\leq (1+\epsilon) e^{-1+1/k} n \ .\end{aligned}$$

If the input is a YES instance, then clearly, the maximum k-unique coverage is n. This is because there exists i such that $i \in S_1 \cap \ldots \cap S_k$ and therefore $V_1^i, \ldots, V_k^i$ are in the stream and these sets uniquely cover all elements.

Therefore, any constant pass algorithm that returns better than a $e^{1-1/k}/(1+\epsilon)$ approximation to `Max Unique Coverage` for some large enough constant success probability implies a protocol to solve $\mathtt{DISJ}(m, k)$. Thus, $\Omega(m/k^2)$ space is required. ◀

5.3 Lower bound for $1+\epsilon$ approximation

Assadi [6] presents a $\Omega(m/\epsilon^2)$ lower bound for the space required to compute a $1+\epsilon$ approximation for `Max Coverage` when $k=2$, even when the stream is in a random order and the algorithm is permitted constant passes. This is proved via a reduction to multiple instances of the Gap-Hamming Distance problem on a hard input distribution, where an input with high maximum coverage corresponds to a YES answer for some Gap-Hamming Distance instance, and a low maximum coverage corresponds to a NO answer for all GHD instances. This hard distribution has the additional property that high maximum coverage inputs also have high maximum unique coverage, and low maximum coverage inputs have low maximum unique coverage. Therefore, the following corollary holds:

▶ **Corollary 18.** *Any constant-pass randomized algorithm with an approximation factor $1+\epsilon$ for `Max Unique Coverage` requires $\Omega(m/\epsilon^2)$ space.*

6 Handling Insert-Delete Streams

6.1 Proof of Theorem 7

Consider coloring the elements of a universe with a 2-wise hash-function such that each element is equally likely to get one of $c = 10d^2k$ colors.

We say a set has color P if the colors of its elements are all different and form the set P. Then, via ℓ_0 sampling [41], use $\tilde{O}(c^d)$ space to sample a set (if one exists) that is colored P (i.e., for each color in P there is exactly one element in the sampled set with this color) for each subset $P \subseteq \{1, 2, \ldots, c\}$ of size at most d.

▶ **Definition 19.** *Let C be a collection of at most k sets where each set have size at most d. Say a set S in C is* good *with respect to C if the elements of S receive different colors and they are all different from the colors received by elements in $(\cup_{S' \in C} S') \setminus S$.*

For any good set S in the collection, let $r(S)$ be the set found by the sampling algorithm that is colored the same as set S. We call $r(S)$ the *replacement* for S.

▶ **Lemma 20.** *Removing sets $S_1, S_2, \ldots, S_g$ that are good with respect to (w.r.t.) C from C and replacing them by $r(S_1), r(S_2), \ldots, r(S_g)$ yields a new collection that (uniquely) covers at least the same number of elements as C.*

Proof. Let R_0 be the set of colors used to color elements in $\cup_{i=1}^{g} S_i$ and let R_1 be the set of colors used to color elements in $(\cup_{S' \in C} S') \setminus (\cup_{i=1}^{g} S_i)$. Because $S_1, S_2, \ldots, S_g$ are good sets, $|R_0| = |\cup_{i=1}^{g} S_i|$ and $R_0 \cap R_1 = \emptyset$. After replacing $S_1, S_2, \ldots, S_g$ by $r(S_1), r(S_2), \ldots$, the multiplicity of an element with a color in R_1 is unchanged. For any color in R_0, let e be the element in $\cup_{i=1}^{g} S_i$ with this color. There will be at least one element with the same color as e after the collection is transformed. It follows that the coverage of the collection does not decrease: the removal of $S_1, S_2, \ldots, S_g$ reduces the coverage by at most $|\cup_{i=1}^{g} S_i|$ but adding $r(S_1), r(S_2), \ldots$ increases the coverage by at least $|R_0|$. To argue that the unique coverage of the collection does not decrease, note that if e had multiplicity 1 then the element with the same color as e after the transformation also has multiplicity 1.

◀

▶ **Lemma 21.** *For any $C' \subseteq C$, $\Pr[$number of good sets in C' is $\geq 4|C'|/5] \geq 1/2$.*

Proof. First note that, a set is not good if one of its element shares a color with an element in that set or in another set in the collection. By the union bound,

$$\Pr[\text{set is not good}] \leq d(dk)/c = 1/10 \ .$$

Hence, for any subset C' of C, $\mathrm{E}\left[\text{number of bad sets in } C'\right] \leq |C'|/10$ and the lemma follows via Markov inequality. ◀

▶ **Theorem 22.** *After repeating the random coloring and sampling $O(\log k)$ times, we have a collection of sets that includes the collection of size at most k that (uniquely) covers the maximum number of elements.*

Proof. For the sake of analysis, let C_0 be a collection of at most k sets with optimum (unique) coverage. Let $C' = C_0$.

1. Randomly color elements. Let C_1 be the collection formed from C_0 by replacing all sets in C_0 that are good sets wrt C_0 by their replacements. Remove all good sets (w.r.t. C_0) from C'.
2. Randomly color elements. Let C_2 be the collection formed from C_1 by replacing all sets in C' that are good sets wrt C_1 by their replacements. Remove all good sets (w.r.t. C_1) from C'.
3. ... continue in this way for $O(\log k)$ steps.

In each step, the size of $|C'|$ decreases by a constant factor with constant probability by appealing to Lemma 21. Hence after $O(\log k)$ steps $|C'| = 0$. Note that the (unique) coverage of $C_{O(\log k)}$ is at least the (unique) coverage of C_0 by Lemma 20. ◀

Noting that the $O(\log k)$ colorings/sampling can be performed in parallel, we have a single-pass algorithm.

6.2 Handling deletions for the algorithm in Theorem 12

We now explain how the approach using in Theorem 12 can be extended to the case where sets may be inserted and deleted. In this setting, it is not immediately obvious how to select the largest $\lceil rk/\epsilon \rceil$ sets; the approach used when sets are only inserted does not extend. Note that in this model we can set m to be the maximum number of sets that have been inserted and not deleted at any prefix of the stream rather than the total number of sets inserted/deleted.

However, we can extend the result as follows. Suppose the sketch of a set for approximating maximum (unique) coverage requires B bits; recall from Section 2.2 that $B = k\epsilon^{-2}\operatorname{polylog}(n, m)$ suffices. We can encode such a sketch of a set S as an integer $i(S) \in [2^B]$. Suppose we know that exactly $\lceil rk/\epsilon \rceil$ sets have size at least some threshold t. We will remove this assumption shortly. Consider the vector $x \in [N]$ where $N = 2^B$ that is initially 0 and then is updated by a stream of set insertions/deletions as follows:

1. When S is inserted, if $|S| \geq t$, then $x_{i(S)} \leftarrow x_{i(S)} + 1$.
2. When S is deleted, if $|S| \geq t$, then $x_{i(S)} \leftarrow x_{i(S)} - 1$.

At the end of this process $x \in \{0, 1, \ldots, , m\}^{2^B}$, $\ell_1(x) = \lceil rk/\epsilon \rceil$, and reconstruct the sketches of largest ηk sets given x. Unfortunately, storing x explicitly in small space is not possible since, while we are promised that at the end of the stream $\ell_1(x) = \lceil rk/\epsilon \rceil$, during the stream it could be that x is an arbitrary binary string with m one's and this requires $\Omega(m)$ memory to store. To get around this, it is sufficient to maintain a linear sketch of x itself that support sparse recovery. For our purposes, the CountMin Sketch [22] is sufficient although other

approaches are possible. The CountMin Sketch allows x to be reconstructed with probability $1-\delta$ using a sketch of size

$$O(\log N + \lceil rk/\epsilon \rceil \log(\lceil rk/\epsilon \rceil/\delta)\log m) = O(\lceil rk/\epsilon \rceil \epsilon^{-2} \operatorname{polylog}(n,m)) \ .$$

To remove the assumption that we do not know t in advance, we consider values:

$$t_0, t_1, \ldots, t_{\lceil \log_{1+\epsilon} m \rceil} \text{ where } t_i = (1+\epsilon)^i \ .$$

We define vector $x^0, x^1, \ldots \in \{0,1,\ldots,,m\}^{2^B}$ where x^i is only updated when a set of size $\leq t_i$ but $> t_{i-1}$ is inserted/deleted. Then there exists i such that $\leq \lceil rk/\epsilon \rceil$ sets have size $\leq t_{i-1}$ and the sketches of these sets can be reconstructed from $x^0, \ldots, x^{t_{i-1}}$. To ensure we have $\lceil rk/\epsilon \rceil$ sets, we may need some additional sketches corresponding to sets of size $> t_{i-1}$ and $\leq t_i$ but unfortunately there could be m such sets and we are only guaranteed recovery of x^{t_i} when it is sparse. However, if this is indeed the case we can still recover enough entries of x^{t_1} by first subsampling the entries at the appropriate rate (we can guess sampling rate $1, 1/2, 1/2^2, \ldots 1/m$) in the standard way. Note that we can keep track of $\ell_1(x^i)$ exactly for each i using $O(\log m)$ space.

7 The Subsampling Framework

Assuming we have v such that $\mathrm{OPT}/2 \leq v \leq \mathrm{OPT}$. Let $h : [n] \to \{0,1\}$ be a hash function that is $\Omega(\epsilon^{-2} k \log m)$-wise independent. We run our algorithm on the subsampled universe $U' = \{u \in U : h(u) = 1\}$. Furthermore, let

$$\Pr[h(u) = 1] = p = \frac{ck \log m}{\epsilon^2 v}$$

where c is some sufficiently large constant. Let $S' = S \cap U'$ and let OPT' be the optimal unique coverage value in the subsampled set system. The following result is from McGregor and Vu [61]. We note that the proof is the same except that the indicator variables now correspond to the events that an element being uniquely covered (instead of being covered).

▶ **Lemma 23.** *With probability at least* $1 - 1/\operatorname{poly}(m)$*, we have that*

$$p\,\mathrm{OPT}(1+\epsilon) \geq \mathrm{OPT}' \geq p\,\mathrm{OPT}(1-\epsilon)$$

Furthermore, if $S_1, \ldots, S_k$ *satisfies* $g(\{S'_1, \ldots, S'_k\}) \geq p\,\mathrm{OPT}(1-\epsilon)/t$ *then*

$$g(\{S_1, \ldots, S_k\}) \geq \mathrm{OPT}(1/t - 2\epsilon) \ .$$

We could guess $v = 1, 2, 4, \ldots, n$. One of the guesses must be between $\mathrm{OPT}/2$ and OPT which means $\mathrm{OPT}' = O(\epsilon^{-2} k \log m)$. Furthermore, if we find a $1/t$ approximation on the subsampled universe, then that corresponds to a $1/t - 2\epsilon$ approximation in the original universe. We note that as long as $v \leq \mathrm{OPT}$ and h is $\Omega(\epsilon^{-2} k \log m)$-wise independent, we have (see [65], Theorem 5):

$$\begin{aligned}&\Pr[g(\{S'_1, \ldots, S'_\ell\}) = p \cdot g(\{S_1, \ldots, S_\ell\}) \pm \epsilon p\,\mathrm{OPT}] \\ &\geq 1 - \exp(-\Omega(k \log m)) \geq 1 - 1/m^{\Omega(k)} \ .\end{aligned}$$

This gives us Lemma 23 even for when $v < \mathrm{OPT}/2$. However, if $v \leq \mathrm{OPT}/2$, then OPT' may be larger than $O(\epsilon^{-2} k \log m)$, and we may use too much memory. To this end, we simply terminate those instantiations. Among the instantiations that are not terminated, we return the solution given by the smallest guess.

References

1 Alexander A. Ageev and Maxim Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Comb. Optim.*, 8(3):307–328, 2004.

2 Shipra Agrawal, Mohammad Shadravan, and Cliff Stein. Submodular secretary problem with shortlists. *CoRR*, abs/1809.05082, 2018. URL: `http://arxiv.org/abs/1809.05082`, `arXiv:1809.05082`.

3 Kook Jin Ahn and Sudipto Guha. Linear programming in the semi-streaming model with application to the maximum matching problem. *Inf. Comput.*, 222:59–79, 2013. URL: `http://dx.doi.org/10.1016/j.ic.2012.10.006`, `doi:10.1016/j.ic.2012.10.006`.

4 Naor Alaluf, Alina Ene, Moran Feldman, Huy L. Nguyen, and Andrew Suh. Optimal streaming algorithms for submodular maximization with cardinality constraints. In *ICALP*, volume 168 of *LIPIcs*, pages 6:1–6:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

5 Aris Anagnostopoulos, Luca Becchetti, Ilaria Bordino, Stefano Leonardi, Ida Mele, and Piotr Sankowski. Stochastic query covering for fast approximate document retrieval. *ACM Trans. Inf. Syst.*, 33(3):11:1–11:35, 2015.

6 Sepehr Assadi. Tight space-approximation tradeoff for the multi-pass streaming set cover problem. In *PODS*, pages 321–335. ACM, 2017.

7 Sepehr Assadi, Sanjeev Khanna, and Yang Li. Tight bounds for single-pass streaming complexity of the set cover problem. In *STOC*, pages 698–711. ACM, 2016.

8 Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: massive data summarization on the fly. In *KDD*, pages 671–680. ACM, 2014.

9 Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, D. Sivakumar, and Luca Trevisan. Counting distinct elements in a data stream. In *RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 2002.

10 Édouard Bonnet, Vangelis Th. Paschos, and Florian Sikora. Parameterized exact and approximation algorithms for maximum k-set cover and related satisfiability problems. *RAIRO Theor. Informatics Appl.*, 50(3):227–240, 2016.

11 Vladimir Braverman, Rafail Ostrovsky, and Dan Vilenchik. How hard is counting triangles in the streaming model? In *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 244–254. Springer, 2013.

12 Marc Bury and Chris Schwiegelshohn. Sublinear estimation of weighted matchings in dynamic data streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 263–274, 2015. URL: `http://dx.doi.org/10.1007/978-3-662-48350-3_23`, `doi:10.1007/978-3-662-48350-3_23`.

13 Amit Chakrabarti and Sagar Kale. Submodular maximization meets streaming: matchings, matroids, and more. *Math. Program.*, 154(1-2):225–247, 2015.

14 Amit Chakrabarti, Subhash Khot, and Xiaodong Sun. Near-optimal lower bounds on the multi-party communication complexity of set disjointness. In *IEEE Conference on Computational Complexity*, pages 107–117. IEEE Computer Society, 2003.

15 Amit Chakrabarti and Anthony Wirth. Incidence geometries and the pass complexity of semi-streaming set cover. In *SODA*, pages 1365–1373. SIAM, 2016.

16 Chandra Chekuri, Shalmoli Gupta, and Kent Quanrud. Streaming algorithms for submodular function maximization. In *ICALP (1)*, volume 9134 of *Lecture Notes in Computer Science*, pages 318–330. Springer, 2015.

17 Rajesh Chitnis and Graham Cormode. Towards a theory of parameterized streaming algorithms. In *14th International Symposium on Parameterized and Exact Computation, IPEC 2019, September 11-13, 2019, Munich, Germany*, pages 7:1–7:15, 2019. URL: `https://doi.org/10.4230/LIPIcs.IPEC.2019.7`, `doi:10.4230/LIPIcs.IPEC.2019.7`.

18 Rajesh Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, Andrew McGregor, Morteza Monemizadeh, and Sofya Vorotnikova. Kernelization via sampling with

applications to finding matchings and related problems in dynamic graph streams. In *SODA*, pages 1326–1344. SIAM, 2016.

19 Rajesh Hemant Chitnis, Graham Cormode, Hossein Esfandiari, MohammadTaghi Hajiaghayi, and Morteza Monemizadeh. Brief announcement: New streaming algorithms for parameterized maximal matching & beyond. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*, pages 56–58, 2015. URL: https://doi.org/10.1145/2755573.2755618, doi:10.1145/2755573.2755618.

20 Rajesh Hemant Chitnis, Graham Cormode, Mohammad Taghi Hajiaghayi, and Morteza Monemizadeh. Parameterized streaming: Maximal matching and vertex cover. In *SODA*, pages 1234–1251. SIAM, 2015.

21 Graham Cormode, Mayur Datar, Piotr Indyk, and S. Muthukrishnan. Comparing data streams using hamming norms (how to zero in). *IEEE Trans. Knowl. Data Eng.*, 15(3):529–540, 2003.

22 Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005. URL: https://doi.org/10.1016/j.jalgor.2003.12.001, doi:10.1016/j.jalgor.2003.12.001.

23 Michael Crouch and Daniel S. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, pages 96–104, 2014. URL: http://dx.doi.org/10.4230/LIPIcs.APPROX-RANDOM.2014.96, doi:10.4230/LIPIcs.APPROX-RANDOM.2014.96.

24 Michael S. Crouch, Andrew McGregor, and Daniel Stubbs. Dynamic graphs in the sliding-window model. In *Algorithms - ESA 2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, pages 337–348, 2013. URL: http://dx.doi.org/10.1007/978-3-642-40450-4_29, doi:10.1007/978-3-642-40450-4_29.

25 Erik D. Demaine, Uriel Feige, MohammadTaghi Hajiaghayi, and Mohammad R. Salavatipour. Combination can be hard: Approximability of the unique coverage problem. *SIAM J. Comput.*, 38(4):1464–1483, 2008.

26 Michael Dom, Jiong Guo, Rolf Niedermeier, and Sebastian Wernicke. Minimum membership set covering and the consecutive ones property. In *SWAT*, volume 4059 of *Lecture Notes in Computer Science*, pages 339–350. Springer, 2006.

27 Yuval Emek and Adi Rosén. Semi-streaming set cover. *ACM Trans. Algorithms*, 13(1):6:1–6:22, 2016.

28 Leah Epstein, Asaf Levin, Julián Mestre, and Danny Segev. Improved approximation guarantees for weighted matching in the semi-streaming model. *SIAM J. Discrete Math.*, 25(3):1251–1265, 2011. URL: http://dx.doi.org/10.1137/100801901, doi:10.1137/100801901.

29 Thomas Erlebach and Erik Jan van Leeuwen. Approximating geometric coverage problems. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008*, pages 1267–1276, 2008. URL: http://dl.acm.org/citation.cfm?id=1347082.1347220.

30 Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

31 Joan Feigenbaum, Sampath Kannan, Andrew McGregor, Siddharth Suri, and Jian Zhang. On graph problems in a semi-streaming model. *Theor. Comput. Sci.*, 348(2):207–216, 2005. doi:http://dx.doi.org/10.1016/j.tcs.2005.09.013.

32 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 *STOC*, pages 1363–1374. ACM, 2020.

33 Daya Ram Gaur, Ramesh Krishnamurti, and Rajeev Kohli. Erratum to: The capacitated max k-cut problem. *Math. Program.*, 126(1):191, 2011.

34 Ashish Goel, Michael Kapralov, and Sanjeev Khanna. On the communication and streaming complexity of maximum bipartite matching. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19,*

2012, pages 468–485, 2012. URL: http://portal.acm.org/citation.cfm?id=2095157&CFID=63838676&CFTOKEN=79617016.

35 Venkatesan Guruswami and Krzysztof Onak. Superlinear lower bounds for multipass graph processing. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, Palo Alto, California, USA, 5-7 June, 2013*, pages 287–298, 2013. URL: http://dx.doi.org/10.1109/CCC.2013.37, doi:10.1109/CCC.2013.37.

36 Sariel Har-Peled, Piotr Indyk, Sepideh Mahabadi, and Ali Vakilian. Towards tight bounds for the streaming set cover problem. In *PODS*, pages 371–383. ACM, 2016.

37 Chien-Chung Huang, Naonori Kakimura, and Yuichi Yoshida. Streaming algorithms for maximizing monotone submodular functions under a knapsack constraint. In *APPROX-RANDOM*, volume 81 of *LIPIcs*, pages 11:1–11:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

38 Piotr Indyk, Sepideh Mahabadi, Ronitt Rubinfeld, Jonathan Ullman, Ali Vakilian, and Anak Yodpinyanee. Fractional set cover in the streaming model. In *APPROX-RANDOM*, volume 81 of *LIPIcs*, pages 12:1–12:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

39 Piotr Indyk and Ali Vakilian. Tight trade-offs for the maximum k-coverage problem in the general streaming model. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, pages 200–217, 2019. URL: https://doi.org/10.1145/3294052.3319691, doi:10.1145/3294052.3319691.

40 Takehiro Ito, Shin-Ichi Nakano, Yoshio Okamoto, Yota Otachi, Ryuhei Uehara, Takeaki Uno, and Yushi Uno. A 4.31-approximation for the geometric unique coverage problem on unit disks. *Theor. Comput. Sci.*, 544:14–31, 2014.

41 Hossein Jowhari, Mert Saglam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *PODS*, pages 49–58. ACM, 2011.

42 John Kallaugher, Andrew McGregor, Eric Price, and Sofya Vorotnikova. The complexity of counting cycles in the adjacency list streaming model. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, pages 119–133, 2019. URL: https://doi.org/10.1145/3294052.3319706, doi:10.1145/3294052.3319706.

43 Michael Kapralov. Better bounds for matchings in the streaming model. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1679–1697, 2013. URL: http://dx.doi.org/10.1137/1.9781611973105.121, doi:10.1137/1.9781611973105.121.

44 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Approximating matching size from random streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 734–751, 2014. URL: http://dx.doi.org/10.1137/1.9781611973402.55, doi:10.1137/1.9781611973402.55.

45 Michael Kapralov, Sanjeev Khanna, and Madhu Sudan. Streaming lower bounds for approximating MAX-CUT. In *SODA*, pages 1263–1282. SIAM, 2015.

46 Michael Kapralov, Sanjeev Khanna, Madhu Sudan, and Ameya Velingker. $(1 + \omega(1))$-approximation to MAX-CUT requires linear space. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1703–1722, 2017. URL: https://doi.org/10.1137/1.9781611974782.112, doi:10.1137/1.9781611974782.112.

47 Michael Kapralov and Dmitry Krachun. An optimal space lower bound for approximating MAX-CUT. *CoRR*, abs/1811.10879, 2018. URL: http://arxiv.org/abs/1811.10879, arXiv:1811.10879.

48 David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015.

49 Christian Konrad. Maximum matching in turnstile streams. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 840–852, 2015. URL: http://dx.doi.org/10.1007/978-3-662-48350-3_70, doi:10.1007/978-3-662-48350-3_70.

50 Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *APPROX-RANDOM*, volume 7408 of *Lecture Notes in Computer Science*, pages 231–242. Springer, 2012.

51 Christian Konrad and Adi Rosén. Approximating semi-matchings in streaming and in two-party communication. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 637–649, 2013. URL: http://dx.doi.org/10.1007/978-3-642-39206-1_54, doi:10.1007/978-3-642-39206-1_54.

52 Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, pages 1650–1654. AAAI Press, 2007.

53 Fabian Kuhn, Pascal von Rickenbach, Roger Wattenhofer, Emo Welzl, and Aaron Zollinger. Interference in cellular networks: The minimum membership set cover problem. In *COCOON*, volume 3595 of *Lecture Notes in Computer Science*, pages 188–198. Springer, 2005.

54 Pasin Manurangsi. A note on max k-vertex cover: Faster fpt-as, smaller approximate kernel and improved approximation. In *2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019, January 8-9, 2019 - San Diego, CA, USA*, pages 15:1–15:21, 2019. URL: https://doi.org/10.4230/OASIcs.SOSA.2019.15, doi:10.4230/OASIcs.SOSA.2019.15.

55 Andrew McGregor. Finding graph matchings in data streams. *APPROX-RANDOM*, pages 170–181, 2005.

56 Andrew McGregor. Graph stream algorithms: a survey. *SIGMOD Record*, 43(1):9–20, 2014.

57 Andrew McGregor and Sofya Vorotnikova. Planar matching in streams revisited. In *APPROX-RANDOM*, volume 60 of *LIPIcs*, pages 17:1–17:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

58 Andrew McGregor and Sofya Vorotnikova. Triangle and four cycle counting in the data stream model. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 445–456, 2020. URL: https://doi.org/10.1145/3375395.3387652, doi:10.1145/3375395.3387652.

59 Andrew McGregor, Sofya Vorotnikova, and Hoa T. Vu. Better algorithms for counting triangles in data streams. In *PODS*, pages 401–411. ACM, 2016.

60 Andrew McGregor and Hoa T. Vu. Better streaming algorithms for the maximum coverage problem. In *ICDT*, volume 68 of *LIPIcs*, pages 22:1–22:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

61 Andrew McGregor and Hoa T. Vu. Better streaming algorithms for the maximum coverage problem. *Theory of Computing Systems*, pages 1–25, 2018.

62 Neeldhara Misra, Hannes Moser, Venkatesh Raman, Saket Saurabh, and Somnath Sikdar. The parameterized complexity of unique coverage and its variants. *Algorithmica*, 65(3):517–544, 2013. URL: https://doi.org/10.1007/s00453-011-9608-0, doi:10.1007/s00453-011-9608-0.

63 Ashkan Norouzi-Fard, Jakub Tarnawski, Slobodan Mitrovic, Amir Zandieh, Aidasadat Mousavifar, and Ola Svensson. Beyond 1/2-approximation for submodular maximization on massive data streams. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 3826–3835. PMLR, 2018.

64 Barna Saha and Lise Getoor. On maximum coverage in the streaming model & application to multi-topic blog-watch. In *SDM*, pages 697–708. SIAM, 2009.

65 Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-hoeffding bounds for applications with limited independence. *SIAM J. Discrete Math.*, 8(2):223–250, 1995.

66 Mariano Zelke. Weighted matching in the semi-streaming model. *Algorithmica*, 62(1-2):1–20, 2012. URL: `http://dx.doi.org/10.1007/s00453-010-9438-5`, `doi:10.1007/s00453-010-9438-5`.