Streaming Submodular Maximization with Differential Privacy

Anamay Chaturvedi [*1] **Huy L. Nguyen** [*1] **Thy Nguyen** [*1]

Abstract

In this work, we study the problem of privately maximizing a submodular function in the streaming setting. Extensive work has been done on privately maximizing submodular functions in the general case when the function depends upon the private data of individuals. However, when the size of the data stream drawn from the domain of the objective function is large or arrives very fast, one must privately optimize the objective within the constraints of the streaming setting. We establish fundamental differentially private baselines for this problem and then derive better trade-offs between privacy and utility for the special case of decomposable submodular functions. A submodular function is decomposable when it can be written as a sum of submodular functions; this structure arises naturally when each summand function models the utility of an individual and the goal is to study the total utility of the whole population as in the well-known Combinatorial Public Projects Problem. Finally, we complement our theoretical analysis with experimental corroboration.

1. Introduction

Consider the task of a service provider that trains machine learning models for a set of users, e.g., platforms such as Amazon Sagemaker and Microsoft Azure. In many cases, collecting features can be costly and the service provider has to select a limited number of features for their models. For instance, given a data set of health attributes of a number of individuals, different users may want to predict the likelihood of different diseases, and a subset of features may be useful for the illnesses of some patients but extraneous for others. A common metric used to measure the utility of a set of features is the mutual information between the chosen set of features and the empirical likelihood of the diseases of interest under the Naive Bayes assumption. This function is known to exhibit a diminishing returns property and is *submodular* (Krause & Guestrin, 2005b).

Definition 1.1 (Submodular functions)**.** Let V be a finite set and f a function mapping from 2^V to $\mathbb{R}$. For all $S \subset V$, we say that $f(S)$ is the *utility* achieved by set S. For every element $e \in V$ and set $S \subset V$, the *marginal utility* of e over S (denoted $f(e|S)$) is the gain in utility provided by e if we add it to the set S; i.e. $f(e|S) = f(\{e\} \cup S) - f(S)$. The function f is called *submodular* if it has the diminishing returns property, i.e., for every pair of subsets $S \subset T \subset V$ and element e not in T, $f(e|T) \leq f(e|S)$. A submodular function is called *monotone* if for all $S \subset T \subset V$, $f(S) \leq f(T)$. We let OPT denote any arbitrary utility-maximizing subset of V, potentially subject to additional constraints.

The problem of maximizing a submodular function subject to constraints on the subsets of V one is allowed to pick occurs across domains like computer science, electrical engineering (Narayanan, 1997), and economics (Dobzinski & Schapira, 2006). In theoretical computer science, submodular maximization is a fundamental instance of *combinatorial optimization*, generalizing problems like Max-Cut and Facility Location (Schrijver et al., 2003). On the other hand, submodular maximization has also been applied concretely to numerous problems in machine learning such as feature selection for classification (Krause & Guestrin, 2005a), influence maximization in social networks (Kempe et al., 2003), document and corpus summarization (Lin & Bilmes, 2011), result diversification in recommender systems (Parambath et al., 2016) and exemplar based clustering (Dueck & Frey, 2007) – see Mitrovic et al. (2017) and Chaturvedi et al. (2021) for more references.

In many of these applications, one must publicly release a solution to a submodular maximization problem in which the information used to perform the optimization is private. Consider the example above where private data of patients is used to compute the utility of different sets of features. The solution depends on private information, and may reveal too much about the private data set. One way this can occur is when a relatively rare feature is picked, adversaries with

[*]Equal contribution [1]Khoury College of Computer Science, Northeastern University, Boston, USA. Correspondence to: Anamay Chaturvedi <chaturvedi.a@northeastern.edu>, Huy L. Nguyen <nguyen.hu@northeastern.edu>, Thy Nguyen <nguyen.thy2@northeastern.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

side-information (such as where the data was collected) can then de-anonymize records in the private data set or deduce who participated in the data collection. Situations like these motivate the problem of *differentially private* submodular maximization, as first considered in Gupta et al. (2010).

Differential privacy (DP) (Dwork et al., 2006b) is a property that when satisfied by an algorithm, allows one to promise strong information-theoretic guarantees limiting how much an adversary might learn about the private data set based on the output of that algorithm run on the data set.

Definition 1.2 (Differential privacy, Dwork et al. (2006b)**).** Let $\mathcal{X}$ be the set of all possible data points. We say that a pair of data sets A, B drawn from $\mathcal{X}$ are *neighbouring* (denoted $A \sim B$) if they differ in at most one data point I, so for instance $A = B \cup \{I\}$. We say that an algorithm $\mathcal{A}$ mapping from data sets derived from $\mathcal{X}$ to some output codomain $\mathcal{Y}$ is (ε, δ)-differentially private for some $\varepsilon, \delta > 0$ if for all pairs of neighbouring data sets A, B and all $Y \subset \mathcal{Y}$,

$$P(\mathcal{A}(A) \in Y) \leq e^{\varepsilon} P(\mathcal{A}(B) \in Y) + \delta.$$

The definition above says that the likelihood of any set of outcomes Y of an (ε, δ)-DP algorithm can vary by at most an e^{ε} multiplicative factor and an additive δ term if we were to add or drop any one point from our data set. For a given choice of *privacy parameters* ε and δ, (ε, δ)-DP is typically achieved by adding appropriately scaled noise to obfuscate sensitive values in the course of the computation, and may result in trading off some of accuracy to achieve the desired level of privacy. Such trade-offs have been shown to be intrinsic to achieving differential privacy for many problems, but a careful accounting for the noise added and the privacy gained can let one significantly improve this trade-off.

In practice, one picks the value of ε to be a small constant that depends on the desired trade-off between privacy and utility. It is typically required that $\delta = o(1/n)$ to avoid the pathological case of completely revealing one person's data and claiming that privacy is achieved.

Although there is an extensive line of work in privately maximizing submodular functions (Gupta et al., 2010; Mitrovic et al., 2017; Chaturvedi et al., 2021; Rafiey & Yoshida, 2020; Sadeghi & Fazel, 2021), as far as we know there is no work on doing so in the *streaming setting*. In this setting, the elements that may be added to the final solution set arrive in a stream such that either the stream length is significantly greater than the disk space available to the optimization algorithm, or the algorithm must make a decision immediately whether to retain this item for possible inclusion in the solution set (at some cost) or to reject it outright. Submodular maximization under streaming has been studied extensively (Gomes & Krause, 2010; Kumar et al., 2013; Badanidiyuru et al., 2014). Most notably, a $(1-\theta)/2$-approximation algorithm, that retains only $O(\frac{k \log k}{\theta})$ many elements, was introduced by Badanidiyuru et al. (2014) for the problem of streaming submodular maximization. This algorithm is near-optimal as it is known that one cannot do better than an approximation factor of $1/2$ (Feldman et al., 2020).

Problem statement: In this work, we consider the problem at the intersection of these two lines of inquiry, i.e. *submodular maximization in the streaming setting under the constraint of differential privacy*. For every possible private data set A there is a corresponding monotonic (non-decreasing) submodular function f_A, a public stream of elements V of length n, and a cardinality constraint k, and we want to find a subset S of V with cardinality at most k that achieves utility close to $f_A(\mathrm{OPT})$ in the streaming setting. Following previous work (Gupta et al., 2010; Mitrovic et al., 2017), we assume a known bound on the *sensitivity* of f_A to A, i.e. for any $A \sim B$, $\max_{S \subset V} |f_A(S) - f_B(S)| \leq \lambda$. Without a bound on the sensitivity, f could be determined completely by a single user and either their privacy or the quality guarantee would have to fail. Even when the sensitivity is at most 1, our lower bound below demonstrates that we are bound to incur non-trivial additive error; it follows that in the case of unbounded sensitivity there can be no (ε, δ)-differentially private algorithm with a non-trivial utility guarantee.

1.1. Contributions

General monotone submodular functions. The starting point for our work is the non-private algorithm of Badanidiyuru et al. (2014) for submodular maximization in the streaming setting, which we would like to privatize. The algorithm is given the submodular function f to be maximized, a cardinality constraint of k, and a guess O for the optimal utility $f(\mathrm{OPT})$. For every stream element $e \in V$, the algorithm adds e to the solution set S if the marginal utility provided by e exceeds the value $O/2k$.

Algorithm 1 Streaming algorithm for monotone submodular maximization subject to a cardinality constraint, Badanidiyuru et al. (2014)

```
input  Monotonic submodular function $f$, cardinality constraint parameter $k$, element stream $V$, approximation parameter $\theta$, estimate $O$ of the optimum cost $f(\mathrm{OPT})$
  $S \leftarrow \emptyset$
  for $e \in V$ do
    if $f(e|S) \geq O/(2k)$ and $|S| < k$ then
      $S \leftarrow S \cup \{e\}$
    end if
  end for
output  $S$
```

The authors showed that this algorithm satisfies the following utility guarantee:

Theorem 1.3 (Badanidiyuru et al. (2014))**.** *The final solution S satisfies $f(S) \geq \min\{O/2, f(\mathrm{OPT}) - O/2\}$.*

We see that if $O = f(\mathrm{OPT})$, then we immediately get a $1/2$-approximation algorithm. More generally if $f(\mathrm{OPT}) \in [E, m]$ for some $E, m \in \mathbb{R}_{>0}$, we can run multiple copies of this algorithm in parallel with a set of geometrically scaling guesses

$$\mathcal{O} = \{E, (1+\theta)E, \ldots, (1+\theta)^{\lfloor \log_{1+\theta} \frac{m}{E} \rfloor} E, m\}.$$

It then follows that there is some $O^* \in \mathcal{O}$ which is within a $(1 \pm \theta)$ multiplicative factor of $f(\mathrm{OPT})$ leading to a net $\frac{1-\theta}{2}$ approximation. Since we must maintain and update a solution set S^O for each $O \in \mathcal{O}$, achieving this guarantee requires that we pay a spatial overhead of $\frac{2 \log m/E}{\theta}$.

One way of privatizing this algorithm is to appeal to the celebrated *sparse vector technique*. Given a sequence of sensitive yes/no queries (as is the case here where either an element is added or not added), the sparse vector technique provides a way of privatizing this sequence of checks with surprisingly little noise and low error. We see that the value $f(e|S)$ is the only private quantity accessed in the submodular streaming maximization algorithm. The sparse vector technique adds independently sampled noise values drawn from a Laplace distribution with standard deviation $\tilde{O}(\sqrt{k}/\varepsilon)$ to the value of $f(e|S)$ and $O/2k$ before making the threshold check. This can be justified by showing that the net privacy lost scales only with number of elements added to the solution set (i.e. at most k). Formalizing this outline leads to the following result:

Theorem 1.4. *Given query access to a monotone submodular function f_A with sensitivity λ taking values in $[0, m]$, and an input stream V of length n, there exists an algorithm that when given a cardinality constraint of k, an approximation parameter θ, a failure probability η, and privacy parameters $\varepsilon < 1$ and δ, is (ε, δ)-DP and achieves utility at least*

$$\frac{(1-\theta) f(\mathrm{OPT})}{2} - O\left(\frac{\lambda k^{1.5} \log^{1.5} \frac{nk}{\eta\theta\delta} \log \frac{m}{E}}{\varepsilon\sqrt{\theta}}\right).$$

with probability $1 - \eta$, where OPT is any arbitrary optimal solution for the non-private submodular maximization problem and $E = \min\{\frac{k \log n}{\varepsilon}, m/2\}$ and the total number of elements retained in memory is $O(\frac{k \log m/E}{\theta})$. Further, this algorithm operates in just one pass.

This result serves as a differentially private baseline for general monotone submodular functions - it achieves a similar multiplicative approximation factor as the non-private algorithm, but suffers significant additive loss. Minimizing this additive loss is key to achieving better utility for private streaming submodular maximization. Are there reasonable assumptions under which we can achieve even lower additive loss whilst preserving privacy?

Decomposable monotone submodular functions. In the non-streaming setting, Gupta et al. (2010) showed that when the private submodular function to be maximized is λ-*decomposable*, i.e., it can be written as a sum of monotonic submodular functions taking values in $[0, \lambda]$ each corresponding to the data of one individual, then a much smaller amount of noise needs to be added for DP than in the general case.

$$f_A(S) = \sum_{p \in A} f_{\{p\}}(S)$$

For each agent p, the function $f_{\{p\}}$ is monotone submodular and takes values in $[0, \lambda]$, which automatically implies f_A has sensitivity λ. Gupta et al. (2010) achieved their result by picking k elements in sequence that approximately maximize the marginal utility that they return. To preserve privacy, these picks are made through the exponential mechanism, which randomly selects an element with probability proportional to its marginal utility. The authors performed an intricate analysis to show that the net privacy loss is in fact independent of the number of elements picked.

In the streaming setting, however, the exponential mechanism cannot be applied as the algorithm only has access to the marginal utility of the current element in the stream at any time whereas executing the exponential mechanism requires knowledge of the marginal utility values of all elements (in particular elements that have not yet been seen by the algorithm). This limited access to the marginal utility values make the previous approach inapplicable, and therefore leads us to investigate the following question:

Is possible to achieve a better utility in the streaming setting for monotone submodular functions that are decomposable?

Our main result answers this question in the affirmative. To achieve this result we move beyond prior work in streaming with privacy and show that even when picking arbitrarily many elements k from an arbitrarily long stream of length n, remarkably it still suffices to add privatizing noise with just *constant* magnitude to the threshold check for each element. The core algorithmic change made is that we sample privatizing noise from the Gumbel distribution instead of the Laplace distribution. Although the Gumbel distribution has been used before in DP to simulate the exponential mechanism, its use in this manner to achieve privacy in the streaming setting is as far as we know entirely novel.

Theorem 1.5. *Given query access to a λ-decomposable monotone submodular function f_A with m summands over a stream V of length n, there exists an algorithm that when given a cardinality constraint of k, an approximation parameter θ, a failure probability η, and privacy parameters*

$\varepsilon < 1$ and δ, achieves utility at least

$$\frac{(1-\theta)f(\mathrm{OPT})}{2} - O\left(\frac{\lambda k \log^2 \frac{m}{E\varepsilon\delta\theta} \log \frac{2nk\log m/E}{\eta\theta}}{\varepsilon\sqrt{\theta}}\right).$$

with probability $1-\eta$. Here $E = \min\{\frac{k\log n}{\varepsilon}, m/2\}$ and the total number of elements retained in memory is $O(\frac{k\log m/E}{\theta})$. Further, this algorithm operates in just one pass.

The privacy analysis of this result turns out to be quite involved and we give more detail below. The multiplicative approximation factor equals the $(1-\theta)/2$-approximation factor of the non-private setting. However, similar to the private non-streaming setting in Chaturvedi et al. (2021) with matroid constraints as well as non-monotone objectives, we see a trade-off between the multiplicative and additive terms which we can tune via the multiplicative approximation parameter θ.

Near-optimality. To show that the additive error term has the optimal dependence on k/ε (up to logarithmic terms) we also extend previous lower bounds by Gupta et al. (2010) for private submodular maximization from the $(\varepsilon, 0)$ to (ε, δ)-setting. The proof proceeds similarly to that of the lower bound of Nguyen et al. (2021) for k-means clustering in the (ε, δ) setting, and the formal statement is as follows:

Theorem 1.6. *For all $0 \leq \varepsilon, \delta \leq 1$, $k \in \mathbb{N}$, $n \geq k\frac{e^{\varepsilon}-1}{\delta}$, and $c \geq \frac{4\delta}{e^{\varepsilon}-1}$, if an (ε, δ)-DP algorithm for the submodular maximization problem for decomposable objectives achieves a multiplicative approximation factor of c, it must incur additive error $\Omega((kc/\varepsilon)\log(\varepsilon/\delta))$.*

The gap in the exponent of k between the upper bound on the additive loss for general λ-sensitive submodular functions and the lower bound occurs in the non-streaming setting as well (compare Mitrovic et al. (2017) and our lower bound Theorem 1.6), and the assumption under which this gap is closed is the same as ours - i.e. λ-decomposability. Further, as we have observed before, the multiplicative approximation factor that we achieve is essentially the same as the non-private setting. In this sense our results are tight with the state of the art for both general private submodular maximization and streaming submodular maximization.

We also find an improvement for decomposable submodular functions by using Gumbel noise instead of Laplace noise in practice by conducting experiments comparing their performance. We include all complete proofs and omitted technical details in the appendix.

1.2. Related work

Papadimitriou et al. (2008) introduced the Combinatorial Public Projects problem (CPPP) - given a set A of m agents and n resources, and a private submodular utility function $f_{\{p\}}$ for every agent p the solver must coordinate with the agents to maximize $f_A := \sum_{p\in A} f_{\{p\}}$, i.e. the sum of their utilities. This problem captures public welfare maximization and is interesting theoretically because in this setting agents are incentivized to lie to the solver and over-represent the utility they may individually derive from the resources that are picked for the group. The authors showed that unless $NP \subset BPP$, there is no truthful and computationally efficient algorithm for the solver to achieve an approximation ratio better than $n^{1/2-\epsilon}$ for any $\epsilon > 0$.

Gupta et al. (2010) were the first to consider the problem of differentially private submodular maximization. They showed that it is possible to privately optimize the objective in CPPP while losing an amount of privacy that is *independent* of k, the number of items picked, and achieved essentially optimal additive error. Since (ε, δ) privacy implies approximate $(2\varepsilon + \delta)$-truthfulness, their result showed that a slight relaxation of the truthfulness condition considered by Papadimitriou et al. (2008) allowed constant factor approximation if the optimal utility was not too low. They also showed that optimizing a submodular function ε-privately under a cardinality constraint of k over a ground set of n elements must suffer additive loss $\Omega(k \log n/k)$.

Mitrovic et al. (2017) considered the more general case of private submodular maximization when the objective function has bounded sensitivity. They were motivated by the problem of feature selection under the constraint of differential privacy. They proposed algorithms for both general monotone and non-monotone objectives with bounded sensitivity, and provided extensions to matroid and p-extendable systems constraints. Although the error guarantees of their results match those of Gupta et al. (2010) in the case of decomposable functions, for the case of general monotone and non-monotone objectives they get higher additive error.

Chaturvedi et al. (2021) extended the results of Gupta et al. (2010) from cardinality to matroid constraints, and from monotone to the non-monotone setting. They achieved this by adapting the Continuous Greedy algorithm of Călinescu et al. (2011) and the Measured Continuous Greedy algorithm of Feldman et al. (2011). They also made a small fix to the privacy analysis of Gupta et al. (2010) resulting in a weaker bound (by a constant factor) on the privacy loss.

Rafiey & Yoshida (2020) also studied the problem of private submodular and k-submodular maximization subject to matroid constraints, and achieved the same multiplicative approximation factor as Chaturvedi et al. (2021) for monotone submodular maximization. In this work privacy and time-efficiency were optimized at the cost of higher additive error.

Sadeghi & Fazel (2021) made further progress on private monotone submodular maximization for submodular functions with *total curvature* at most $\kappa \in [0, 1]$ by achieving a $(1 - \kappa/e)$-approximation algorithm and lower query complexity than the previous works.

Salazar & Cummings (2021) considered a variant of this line of work wherein a sequence of private submodular functions defined over a common public ground set are processed, and at every iteration a set of at most k elements from the ground set must be picked before the function is observed. Here the goal is *regret minimization*, and the authors introduced differentially private algorithms that achieve sub-linear regret with respect to a $(1 - 1/e)$-approximation factor in the full information and bandit settings.

2. Preliminaries

We first make a couple of simplifying technical observations.

Remark 2.1. Following the setup of the DP set cover problem in Gupta et al. (2010), we assume that there is a publicly known upper bound on the number of agents which we denote by m; as the dependence of the error and space on m will be seen to be logarithmic, even a loose upper bound works well. Alternatively, we can allocate a small fraction of our privacy budget to privatize the number of agents m via the Laplace mechanism (Lemma 2.6).

Remark 2.2. It will suffice to reason about functions with sensitivity 1, and 1-decomposable functions. For the more general case where the submodular function f has sensitivity λ or is λ-decomposable, we can reason about f/λ which ensures that our assumptions hold.

We will use the following *basic* and *advanced composition* laws to reason about the privacy loss that occurs when multiple differentially private mechanisms are used in a modular fashion as subroutines. We follow the formulation in Dwork & Roth (2014).

Theorem 2.3 (Basic composition, Dwork et al. (2006a); Dwork & Lei (2009))**.** *If $\mathcal{M}_i$ is $(\varepsilon_i, \delta_i)$-differentially private for $i = 1, \dots, k$, then the release of the outputs of all k mechanisms is $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private.*

Theorem 2.4 (Advanced composition, Dwork et al. (2010))**.** *For all $\varepsilon, \delta, \delta' > 0$, given a set of (ε, δ)-differentially private mechanisms, if an adversary adaptively chooses k mechanisms to run on a private data set, then the tuple of responses is $(\varepsilon', k\delta + \delta')$-differentially private for $\varepsilon' = \sqrt{2k \ln(1/\delta')}\varepsilon + k\varepsilon(e^{\varepsilon} - 1)$. In particular, for $\varepsilon' < 1$, if $\varepsilon = \frac{\varepsilon'}{2\sqrt{2k \ln(1/\delta)}}$, then the net k-fold adaptive release is $(\varepsilon, (k+1)\delta)$-differentially private.*

We will appeal to the following private mechanism used to choose an element from a public set based on a private score function of bounded sensitivity.

Lemma 2.5 (Exponential Mechanism, McSherry & Talwar (2007))**.** *Let $q : \mathcal{X}^* \times V \to \mathbb{R}$ be a score function for a public domain V that depends on the private input data set drawn from $\mathcal{X}^*$, i.e. $q(A, v)$ is the score of $v \in V$ for the data set $A \in \mathcal{X}^*$. Let $\Delta q = \max_{A \sim B} \max_{v \in V} |q(A, v) - q(B, v)|$ be the sensitivity of q, i.e. the maximum possible change in the value of the score of an element for neighboring data sets. The exponential mechanism $\mathcal{M}$ samples $v \in V$ with probability $\propto \exp(\varepsilon q(A, v)/(2\Delta q))$ and outputs v^*. The exponential mechanism is ε-differentially private. Further, for a finite set V, we have that with probability $1 - \gamma$,*

$$q_A(v^*) \geq \max_{v \in V} q_A(v) - \frac{2\Delta q}{\varepsilon} \ln \frac{|V|}{\gamma}.$$

Given a real-valued function with bounded sensitivity, the Laplace mechanism can be used to privatize the value taken by that function on the private input data set.

Lemma 2.6 (Laplace mechanism, Dwork et al. (2006b))**.** *Given a function $f : 2^{\mathcal{X}} \to \mathbb{R}$ such that $\max_{A \sim B} |f(A) - f(B)| \leq \Delta f$, the mapping $A \mapsto f(A) + \alpha$ where $\alpha \sim \mathrm{Lap}(\Delta f/\varepsilon)$ is ε-differentially private. Here $\mathrm{Lap}(\sigma)$ denotes the standard Laplace distribution with scale parameter σ.*

We will also draw random values from the Gumbel distribution for improved privacy guarantees. We recall the distribution function for later use.

Definition 2.7 (Gumbel distribution)**.** The Gumbel distribution is defined on $\mathbb{R}$. When the mean is μ and the scale parameter is γ, the distribution is defined by its CDF

$$F(x) = \exp\Big(-e^{-(x-\mu)/\gamma}\Big),$$

or alternatively its density function

$$f(x) = \frac{1}{\gamma} \exp\Big(-(x - \mu)/\gamma + e^{-(x-\mu)/\gamma}\Big).$$

2.1. The sparse vector technique

Following Dwork & Roth (2014), we recall the pseudocode and utility guarantee of the sparse vector technique. This result is attributed to Dwork et al. (2009); we refer the reader to the end of chapter 3 of Dwork & Roth (2014) for a more comprehensive discussion.

Theorem 2.8 (Dwork et al. (2009))**.** *For $\sigma = (\sqrt{32k \ln \frac{1}{\delta}})/\varepsilon$, $\mathcal{D}_\alpha = Lap(\sigma)$ and $\mathcal{D}_\beta = Lap(2\sigma)$, Algorithm 2 is (ε, δ)-DP. Further, with probability $1 - \beta$, for all queries i, if $\hat{\nu}_i$ is the privatized query value $f_i + \nu_i$, then*

$$|\hat{\nu}_i - \nu_i| \leq \frac{(\ln n + \ln \frac{4c}{\beta})\sqrt{k \ln \frac{2}{\delta}}(\sqrt{512} + 1)}{\varepsilon}.$$

Algorithm 2 Sparse, Dwork et al. (2009); Dwork & Roth (2014)

```
input  Arbitrary (possibly adaptive) stream of sensitivity 1
    queries $f_1, f_2, \dots$, a threshold $T$, a cutoff point $k$, thresh-
    old noise $\mathcal{D}_\alpha$, score noise $\mathcal{D}_\beta$.
  $\alpha \sim \mathcal{D}_\alpha$
  Let $\hat{T}_i = T + \alpha$ for $i \in \{0, \dots, k-1\}$
  Let count $= 0$
  for query $e_i$ do
    Let $\beta_i \sim \mathcal{D}_\beta$
    if $f_i + \beta_i \geq \hat{T}_{\text{count}}$ then
      $a_i = \top$
      count $=$ count $+ 1$
    else
      $a_i = \bot$
    end if
    if count $\geq k$ then
      Halt
    end if
  end for
output  Stream of yes/no outputs $(a_1, a_2, \dots)$
```

3. Private streaming with Laplace noise

Our main algorithm for maximizing submodular monotone functions in both the general and decomposable cases is Algorithm 3. At a high level we would like to adapt the non-private submodular streaming algorithm of Badanidiyuru et al. (2014) to the private setting by privatizing the check that is made when adding an element e to the solution set S. Doing so essentially gives us an instantiation of Algorithm 2 wherein the sensitivity 1 queries are the values $f(e|S)$, the marginal utilities of the stream elements for the private function over the solution set picked thus far, and the threshold being compared with is $O/2k$, where O is a guess for the optimal utility OPT.

In the case where f is monotone submodular and has sensitivity 1 but is not decomposable, Algorithm 3 sets the privatizing noise distributions $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ directly according to the analysis of the sparse vector technique, i.e. Theorem 2.8.

As in the non-private case, since the value OPT is not known, we let O vary over a set $\mathcal{O} = \{E, (1+\theta)E, \dots, m\}$; it follows that $|\mathcal{O}| = \lceil \log_{1+\theta} \frac{m}{E} \rceil + 1$, this is denoted T in the pseudo-code. In this case the ideal choice of E is the lowest possible additive error that we can achieve in the DP setting, and can be set to be equal to the lower bound which we derive in Theorem 1.6. Indeed, if $f(\text{OPT}) < E$, it can be shown that the utility guarantees hold trivially. Further, we observe that our net privacy budget of (ε, δ) must be split across these T-many calls to Algorithm 2 by the composition laws of privacy described in the preliminary (for a stronger theoretical guarantee one appeals to the advanced composition rule); this is how we derive our expression for ε' which is the privacy parameter passed in each call to Algorithm 2. The stream V can then be processed in parallel for each guess $O \in \mathcal{O}$. At the end of the stream we see that we have some T-many solutions S^O corresponding to different guesses $O \in \mathcal{O}$ for OPT. To choose a final solution one again has to access the private values $f_A(S^O)$, and to account for this we appeal to the exponential mechanism to choose a candidate S^{O^*} that achieves near-maximal utility among the set of guesses $\{S^O : \mathcal{O}\}$. Applying the composition laws for privacy across all calls to private mechanisms and accounting for the additive error introduced by appealing to the sparse vector technique one derives Theorem 1.4.

Algorithm 3 Private streaming submodular maximization

```
input  Monotone submodular function $f_A$, cardinality con-
    straint parameter $k$, failure probability $\beta$, privacy parame-
    ters $(\varepsilon, \delta)$, element stream $V$, approximation parameter
    $\theta$
  Let $E = \min\left\{\frac{k \log n}{\varepsilon}, m/2\right\}$
  Let $T = \lceil \log_{1+\theta} \frac{m}{E} \rceil + 1$
  if $f_A$ has sensitivity 1 but is not 1-decomposable then
    Let $\varepsilon' = \frac{\varepsilon}{4\sqrt{2T \ln((T+1)/\delta)}}$
    Let $\sigma = \frac{\sqrt{32k \ln \frac{T+1}{\delta}}}{\varepsilon'}$
    Let $\mathcal{D}_\alpha = \text{Lap}(\sigma)$
    Let $\mathcal{D}_\beta = \text{Lap}(2\sigma)$
  else if $f_A$ is 1-decomposable then
    $\gamma = O\left(\frac{\varepsilon}{\sqrt{T} \log^{1.5} T/\delta}\right)$ (exact expression in
    Lemma 4.2)
    $\mathcal{D}_\alpha = \text{Gumb}(\gamma)$
    $\mathcal{D}_\beta = \text{Gumb}(\gamma)$
  end if
  Let $\mathcal{O} = \{E, (1+\theta)E, (1+\theta)^2 E, \dots, (1+
  \theta)^{\lfloor \log_{1+\theta} \frac{m}{E} \rfloor} E, m\}$
  for $e_i \in V$ do
    for all $O \in \mathcal{O}$ do
      $S^O \leftarrow$ Algorithm 2
      (Query stream $(f_A(e|S^O))_{e \in V}$, threshold $\frac{O}{2k}$,
      cutoff $k, \mathcal{D}_\alpha, \mathcal{D}_\beta$)
      where

          $a_i = \top \Rightarrow$ add element $e_i$ to $S^O$
          $a_i = \bot \Rightarrow$ reject element $e_i$

    end for
  end for
  $S^{O^*} \leftarrow$ Exponential Mechanism($\{S^O : O \in \mathcal{O}\}$, $f_A(\cdot)$,
  privacy parameter $\varepsilon/2$) (in the notation of Lemma 2.5,
  $q(A, S^O) = f_A(S^O)$)
output  $S^{O^*}$
```

4. Private streaming with Gumbel Noise

In the setting where the given objective is decomposable, we show that by setting our privatizing noise distributions $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ equal to a Gumbel distribution with an appropriate scale parameter, we can reduce the dependence of the additive error on k/ε to what is asymptotically the best possible. As the utility analysis in this case is identical to that before, we focus on just the privacy analysis in this section.

We fix some arbitrary guess O for OPT. For any possible output set $S = \{e_{i_1}, \dots, e_{i_k}\}$, we want to bound the ratio $\Pr[\mathcal{A}(f_A) = S] / \Pr[\mathcal{A}(f_B) = S]$, where $A = B \cup I$ (the other case $B = A \cup I$ turns out to be relatively straightforward). The output $S = \{v_{i_1}, \dots, v_{i_k}\}$ is equivalent to a stream of outputs $a_1, a_2, \dots a_{|V|}$, where $a_i = \top$ denotes that the element e_i was picked, and $a_i = \bot$ denotes rejection. For every index i, let S_i denote the set of elements already accepted when the element e_i is being processed. We have the following technical lemma:

Lemma 4.1. *Conditioned on having picked the set S_i of elements with $|S_i| = u < k$,*

$$\Pr[a_i = \top | S_i] = \int_{-\infty}^{\infty} 1 - \exp\Big(-w_A(i|S_i) e^{\alpha_u/\gamma}\Big) \mathrm{d}\alpha_{|S_i|},$$

where

$$w_A(i|S_i) = \exp\bigg(\frac{-1}{\gamma}\left(\frac{O}{2k} - f_A(e_i|S_i)\right)\bigg),$$

and $\alpha_u \sim Gumbel(0, \gamma)$.

We think of the term $w_A(i|S_i)$ as the *weight* of element e_i; we see that the probability that e_i is accepted increases as its weight increases. By appealing to this lemma iteratively and integrating over the noise variables α_u, it is possible to show that $\Pr[\mathcal{A}(A) = S] = \prod_{u=1}^{r} X_u$ where X_u equals

$$\frac{w_A(i_u|S_{i_u})}{\bigg(1 + \sum\limits_{j \in (i_{u-1}, i_u)} w_A(j|S_{i_u})\bigg)\bigg(1 + \sum\limits_{j \in (i_{u-1}, i_u]} w_A(j|S_{i_u})\bigg)}.$$

From some elementary algebra it then follows that

$$\frac{\Pr[\mathcal{A}(f_A) = S]}{\Pr[\mathcal{A}(f_B) = S]} \le \left(\prod_{u=1}^{r} \frac{1 + \sum_{j \in (i_{u-1}, i_u)} w_A(j|S_{i_u})}{1 + \sum_{j \in (i_{u-1}, i_u)} w_B(j|S_{i_u})}\right)^2.$$

We see that by definition $w_A(j|S_{i_u}) = w_B(j|S_{i_u}) \exp\big(f_{\{I\}}(e_j|S_{i_u})/\gamma\big)$, whence we can write

$$\frac{1 + \sum\limits_{j \in (i_{u-1}, i_u)} w_A(j|S_{i_u})}{1 + \sum\limits_{j \in (i_{u-1}, i_u)} w_B(j|S_{i_u})} = \mathop{\mathrm{E}}_{j \sim P_u}[\exp\big(f_{\{I\}}(e_j|S)/\gamma\big)].$$

Here P_u is the distribution on (i_{u-1}, i_u) that picks element j with probability $\propto w_B(e_j|S_{i_u})$, and no element at all with probability $\propto 1$. At a high level, what we show is that the increase in the likelihood of an element e_{i_u} being picked can be bounded in terms of the expectation of a function of the marginal gain of e_j over S_{i_u} for the agent I when e_j is drawn according to P_u. We denote this expectation term Y_u, whence we can write

$$\frac{\Pr[\mathcal{A}(f_A) = S]}{\Pr[\mathcal{A}(f_B) = S]} \le \left(\prod_{u=1}^{r} Y_u\right)^2.$$

We use this upper bound to show that for most sets S that are likely to be picked by the algorithm given the function f_B (i.e. with probability $1 - \delta$), $\prod_{u=1}^{k} Y_u$ and consequently the likelihood of S under f_A is not too large. Since the submodular function f_I takes values in $[0, 1]$, intuitively we expect the product of the Y_u to telescope over u and concentrate strongly. We also see that if Y_u is large, then the likelihood of some element with its index in the interval (i_{u-1}, i_u) being picked is relatively high (note that although the distribution according to which the next element is picked and P_u are not identical, they are similar and closely related). In particular, it should be unlikely that the algorithm picks sequence of elements $(e_{i_1}, \dots, e_{i_k})$ for which this product is large.

However, to formally derive a privacy guarantee from this outline there are many technical challenges that need to be resolved. The distribution over which the expectation term of interest depends on the set of elements already picked. We formalize this situation by defining a multi-round probabilistic process and proceeding by induction. The expectation term derived above does not admit a useful concentration bound directly, so instead we must analyze a $\tilde{O}(1/\varepsilon)$th-moment of the net privacy loss. The formal statement that we derive is as follows:

Lemma 4.2. *Consider the following k-round probabilistic process. Let $v_j := w_B(i_{u-1} + j|S_{i_u})$. In each round u, it is the case that the set of elements $S_{i_u} = \{i_1, \dots, i_{u-1}\}$ has been picked, and the element $i_u = j + i_{u-1}$ is picked with probability*

$$p_j = \frac{1}{1 + v_1 + \cdots + v_{j-1}} \cdot \frac{v_j}{1 + v_1 + \cdots + v_j}.$$

Then, for each $q = 1, \dots, r$, for a value of $c = \gamma/4 = \frac{2}{\varepsilon \ln 2} \log \frac{2}{\varepsilon\delta}$, the following bound holds:

$$\mathop{\mathrm{E}}_{S}\left[\prod_{u=q}^{k} Y_u^c | S_{i_q}\right] \le 1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_q})).$$

We see that the right hand side of the bound above has a telescoping form similar to what we expected - the higher the utility of the current set S_{i_q}, the lower is the expected product of the subsequent Y_u terms for $u > q$.

We recall that we bounded from above the ratio of probabilities, $\Pr[\mathcal{A}(f_A) = S]/\Pr[\mathcal{A}(f_B) = S]$ by $\left(\prod_{u=1}^{k} Y_u\right)^2$. We can now apply Markov's inequality on the c-th exponent of this random variable so as to exploit the bound on the expectation that we have derived; this completes the core of our privacy analysis. Similar to before we must account for the T many calls to Algorithm 2 as well as the exponential mechanism and this is identical to the case with Laplace noise.

5. Experiments

In this section, we empirically evaluate our approach on a practical instance of private submodular maximization, k-medians clustering [1]. Given a set of points V, a metric $d : V \times V \to \mathbb{R}$, a private set of demand points $P \subseteq V$, the objective of the k-medians problem is to select a set of points $S \subset V$, $|S| \leq k$ to minimize $\text{cost}(S) = \sum_{p \in P} d(p, S)$, where $d(p, S) = \min_{s \in S} d(p, s)$. An application of his problem is allocating relatively few (k) service centers to be able to reach a large set of clients (P) and ensure that there is at least one service location not too far from most clients; when the clients' locations are private but the service locations are public, this problem requires a differentially private solver. We can optimize this objective by casting it into the following submodular maximization problem: $\max_{S \subset V, |S| \leq k} \sum_{p \in P} 1 - d(p, S)/G$, where G is a normalization constant so that $f_p(S) = 1 - d(p, S)/G \in [0, 1]$. Setting $d(p, \emptyset) = G$, it can be checked that $\text{cost}(S)$ is a monotone decomposable submodular function.

We compare the performance of Algorithm 3 with **Laplace** and **Gumbel** noise on two data sets. First, following Mitrovic et al. (2017); Chaturvedi et al. (2021) we use the Uber data set (FiveThirtyEight, 2019) of Uber cab pick ups in Manhattan for the month of April 2014; the goal is to allocate public waiting locations for Uber cabs so as to serve requests from clients within short distances. Second, we construct a synthetic dataset in $\mathbb{R}^2$ by generating clients P from a mixture of 3 Gaussian distributions, each with identity covariance matrix and mean chosen uniformly at random from a bounding box of [20] × [20]. We sample 15000 points for one Gaussian distribution, and 2500 points for each of the other two. For both settings, we set $d(\cdot, \cdot)$ to be the ℓ_1 or Manhattan distance, i.e. $d(a, b) = |a_1 - b_1| + |a_2 - b_2|$. We set V to be a 50 × 50 2-D grid of points uniformly spanning the rectangular domain.

We compare our two algorithms with an approach that selects k **Random** points from the stream as a differentially private baseline and the **Non-private** algorithm 1. For both data sets, we set $\delta = 1/|P|^{1.5}$ and $\theta = 0.2$. In Figure 3 and Figure 6 we graph the clustering cost versus the cardinality constraint k on the Taxi and Synthetic data sets respectively. We also tabulate the exact numerical values recorded in the appendix. We measure and report the mean and standard deviation of the clustering cost over 20 random runs with varied k and ε.

We set $E = \min(k \log n/\varepsilon, |P|/2)$ for the private algorithms, and $E = \min(\max_{e_i \in V} f(e_i), k \log n/\varepsilon, |P|/2)$ for the non-private algorithm. This guarantees that the number of thresholds used in the non-private algorithm is at least that used in the private algorithms. Instead of using the exponential mechanism to output the solution in algorithm 3, we use the Report Noisy Max mechanism with equivalent privacy guarantee and similar tail bound (see Dwork & Roth (2014)); this avoids potential overflow issues with the exponential mechanism. When the number of elements left in the stream of a non-private instance is less than $k - |S|$, we add the rest of the points to S. This does not affect the theoretical guarantee, but might benefit the algorithm empirically.

Although we apply advanced composition in our theoretical analyses as it asymptotically requires lower noise than basic composition, because of the difference in constant coefficients, basic composition works better for the number of thresholds we need to consider. For this experiment, we apply basic composition and set $\varepsilon' = \varepsilon/T, \delta' = \delta/T$.

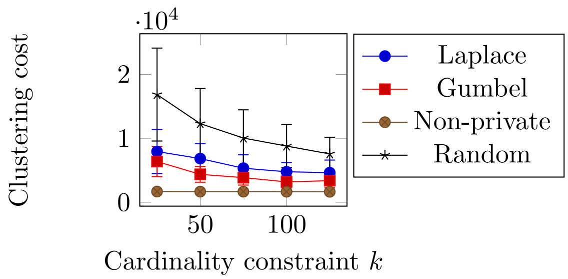
In Figures 1 and 4, we report the results for $\varepsilon = 0.1$. We observe that Gumbel noise outperforms Laplace noise in both settings. We observe similar results in Figures 2 and 5 for $\varepsilon = 1$. Gumbel noise continues to outperform Laplace noise in both settings. Increasing the privacy budget from 0.1 to 1 slightly improves the utility of the differentially private approaches.

One interesting artefact that we observe for the synthetic data set is that the clustering cost actually increases as we increase the solution size from $k = 100$ to $k = 125$ when using Laplace noise. In general this should not happen as increasing the number of centers can only reduce the clustering cost if the centers are picked in the same way across experiments. However, since we have a fixed privacy budget and are forced to split this budget among a greater number of choices when using Laplace noise, we end up making a larger number of poorer quality picks for a net worse solution. This phenomenon has also been seen in other works on private clustering (Nguyen et al., 2021).

On the other hand, since the algorithm with Gumbel noise uses a scale parameter which is invariant in the cardinality constraint, there is no such worsening of performance with a more generous cardinality constraint. We are able to ensure

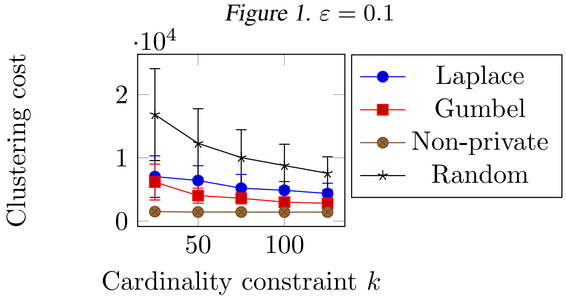
[1] The code used to run all experiments may be found at www.github.com/thydnguyen/PrivSubmodularOpt. All experiments were performed on a PC with 5.2 GHz i9 chip and 64 GB RAM.

that apart from better performance overall, increasing our budget can only lead to a lower clustering cost and we do not need to consider private hyperparameter optimization over k, for instance.



(a)

Figure 1. $\varepsilon = 0.1$



(a)

Figure 2. $\varepsilon = 1$

Figure 3. Comparison of clustering cost (lower is better) achieved by various streaming algorithms on the Taxi data set.

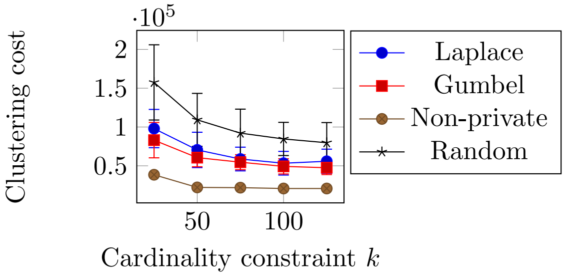
Acknowledgements

All the authors were supported by NSF CAREER grant 1750716.

References

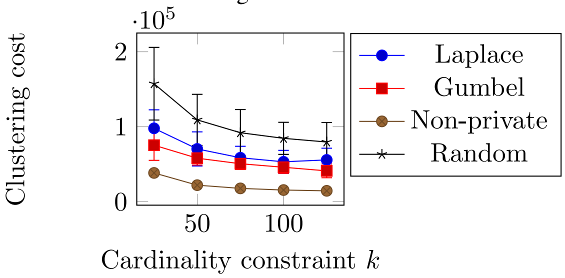
Badanidiyuru, A., Mirzasoleiman, B., Karbasi, A., and Krause, A. Streaming submodular maximization: massive data summarization on the fly. In Macskassy, S. A., Perlich, C., Leskovec, J., Wang, W., and Ghani, R. (eds.), *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pp. 671–680. ACM, 2014. doi: 10.1145/2623330.2623637. URL https://doi.org/10.1145/2623330.2623637.

Călinescu, G., Chekuri, C., Pál, M., and Vondrák, J. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM J. Comput.*, 40(6):1740–1766, 2011. doi: 10.1137/080733991. URL https://doi.org/10.1137/080733991.



(a)

Figure 4. $\varepsilon = 0.1$



(a)

Figure 5. $\varepsilon = 1$

Figure 6. Comparison of clustering cost (lower is better) achieved by various streaming algorithms on the synthetic data set.

Chaturvedi, A., Le Nguyen, H., and Zakynthinou, L. Differentially private decomposable submodular maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35 No. 8., pp. 6984–6992, 2021.

Dobzinski, S. and Schapira, M. An improved approximation algorithm for combinatorial auctions with submodular bidders. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pp. 1064–1073. ACM Press, 2006. URL http://dl.acm.org/citation.cfm?id=1109557.1109675.

Dueck, D. and Frey, B. J. Non-metric affinity propagation for unsupervised image categorization. In *IEEE 11th International Conference on Computer Vision, ICCV 2007, Rio de Janeiro, Brazil, October 14-20, 2007*, pp. 1–8. IEEE Computer Society, 2007. doi: 10.1109/ICCV.2007.4408853. URL https://doi.org/10.1109/ICCV.2007.4408853.

Dwork, C. and Lei, J. Differential privacy and robust statistics. In Mitzenmacher, M. (ed.), *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pp. 371–380. ACM, 2009. doi: 10.

1145/1536414.1536466. URL https://doi.org/10.1145/1536414.1536466.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In Vaudenay, S. (ed.), *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pp. 486–503. Springer, 2006a. doi: 10.1007/11761679_29. URL https://doi.org/10.1007/11761679_29.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. D. Calibrating noise to sensitivity in private data analysis. In Halevi, S. and Rabin, T. (eds.), *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pp. 265–284. Springer, 2006b. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

Dwork, C., Naor, M., Reingold, O., Rothblum, G. N., and Vadhan, S. P. On the complexity of differentially private data release: efficient algorithms and hardness results. In Mitzenmacher, M. (ed.), *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pp. 381–390. ACM, 2009. doi: 10.1145/1536414.1536467. URL https://doi.org/10.1145/1536414.1536467.

Dwork, C., Rothblum, G. N., and Vadhan, S. P. Boosting and differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pp. 51–60. IEEE Computer Society, 2010. doi: 10.1109/FOCS.2010.12. URL https://doi.org/10.1109/FOCS.2010.12.

Feldman, M., Naor, J., and Schwartz, R. A unified continuous greedy algorithm for submodular maximization. In Ostrovsky, R. (ed.), *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pp. 570–579. IEEE Computer Society, 2011. doi: 10.1109/FOCS.2011.46. URL https://doi.org/10.1109/FOCS.2011.46.

Feldman, M., Norouzi-Fard, A., Svensson, O., and Zenklusen, R. The one-way communication complexity of submodular maximization with applications to streaming and robustness. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1363–1374, 2020.

FiveThirtyEight. Uber pickups in new york city, Nov 2019. URL https://www.kaggle.com/datasets/fivethirtyeight/uber-pickups-in-new-york-city.

Gomes, R. and Krause, A. Budgeted nonparametric learning from data streams. In Fürnkranz, J. and Joachims, T. (eds.), *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pp. 391–398. Omnipress, 2010. URL https://icml.cc/Conferences/2010/papers/433.pdf.

Gupta, A., Ligett, K., McSherry, F., Roth, A., and Talwar, K. Differentially private combinatorial optimization. In Charikar, M. (ed.), *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pp. 1106–1125. SIAM, 2010. doi: 10.1137/1.9781611973075.90. URL https://doi.org/10.1137/1.9781611973075.90.

Kempe, D., Kleinberg, J. M., and Tardos, É. Maximizing the spread of influence through a social network. In Getoor, L., Senator, T. E., Domingos, P. M., and Faloutsos, C. (eds.), *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 24 - 27, 2003*, pp. 137–146. ACM, 2003. doi: 10.1145/956750.956769. URL https://doi.org/10.1145/956750.956769.

Krause, A. and Guestrin, C. Near-optimal nonmyopic value of information in graphical models. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pp. 324–331. AUAI Press, 2005a. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1238&proceeding_id=21.

Krause, A. and Guestrin, C. Near-optimal nonmyopic value of information in graphical models. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pp. 324–331. AUAI Press, 2005b. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1238&proceeding_id=21.

Kumar, R., Moseley, B., Vassilvitskii, S., and Vattani, A. Fast greedy algorithms in mapreduce and streaming. In Blelloch, G. E. and Vöcking, B. (eds.), *25th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA '13, Montreal, QC, Canada - July 23 - 25, 2013*, pp. 1–10. ACM, 2013. doi: 10.1145/2486159.2486168. URL https://doi.org/10.1145/2486159.2486168.

Lin, H. and Bilmes, J. A. A class of submodular functions for document summarization. In Lin, D., Matsumoto, Y., and Mihalcea, R. (eds.), *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pp. 510–520. The Association for Computer Linguistics, 2011. URL https://aclanthology.org/P11-1052/.

McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pp. 94–103. IEEE Computer Society, 2007. doi: 10.1109/FOCS.2007.41. URL https://doi.org/10.1109/FOCS.2007.41.

Mitrovic, M., Bun, M., Krause, A., and Karbasi, A. Differentially private submodular maximization: Data summarization in disguise. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2478–2487. PMLR, 2017. URL http://proceedings.mlr.press/v70/mitrovic17a.html.

Narayanan, H. *Submodular functions and electrical networks*, volume 54. Elsevier, 1997.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Nguyen, H. L., Chaturvedi, A., and Xu, E. Z. Differentially private k-means via exponential mechanism and max cover. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):9101–9108, May 2021. doi: 10.1609/aaai.v35i10.17099. URL https://ojs.aaai.org/index.php/AAAI/article/view/17099.

Papadimitriou, C. H., Schapira, M., and Singer, Y. On the hardness of being truthful. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pp. 250–259. IEEE Computer Society, 2008. doi: 10.1109/FOCS.2008.54. URL https://doi.org/10.1109/FOCS.2008.54.

Parambath, S. P., Usunier, N., and Grandvalet, Y. A coverage-based approach to recommendation diversity on similarity graph. In Sen, S., Geyer, W., Freyne, J., and Castells, P. (eds.), *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, pp. 15–22. ACM, 2016. doi: 10.1145/2959100.2959149. URL https://doi.org/10.1145/2959100.2959149.

Rafiey, A. and Yoshida, Y. Fast and private submodular and k-submodular functions maximization with matroid constraints. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7887–7897. PMLR, 2020. URL http://proceedings.mlr.press/v119/rafiey20a.html.

Sadeghi, O. and Fazel, M. Differentially private monotone submodular maximization under matroid and knapsack constraints. In Banerjee, A. and Fukumizu, K. (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2908–2916. PMLR, 2021. URL http://proceedings.mlr.press/v130/sadeghi21a.html.

Salazar, S. P. and Cummings, R. Differentially private online submodular maximization. In *International Conference on Artificial Intelligence and Statistics*, pp. 1279–1287. PMLR, 2021.

Schrijver, A. et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

A. Proof of Theorem 1.3

In this section we reproduce the proof of utility of Algorithm 1 of Badanidiyuru et al. (2014).

Theorem 1.3 (Badanidiyuru et al. (2014))**.** *The final solution S satisfies $f(S) \geq \min\{O/2, f(\mathrm{OPT}) - O/2\}$.*

Proof. We see from the pseudocode of Algorithm 1 that when an element $e_i \in V$ is processed by the stream, if S_i is the set of accepted elements at that point then e_i is retained and added to the solution if and only if $f(e_i|S_i) \geq O/2k$.

Let $S = \{e_{i_1}, \dots, e_{i_{|S|}}\}$. We consider two cases depending on the size of S. If $|S| = k$, then

$$\begin{aligned} f(S) &= \sum_{j=1}^{k} f(e_{i_j}|S_{i_j}) \\ &\geq k \cdot \frac{O}{2k} \\ &\geq O/2. \end{aligned}$$

In the second case, if $|S| < k$, then all elements $e_i \in OPT \setminus S$ must have been rejected i.e. $f(e_i|S_i) < O/(2k)$. Let $OPT \backslash S = \{e_{j_1}, \dots, e_{j_t}\}$ (i.e. $\{j_1, \dots, j_t\}$ is some subset of $\{i_1, \dots, i_{|S|}\}$). Since f is monotonic and $S_i \subset S$ for all $e_i \in V$, we have that $f(e_{j_i}|S) \leq f(e_{j_i}|S_{j_i}) < O/(2k)$. We can write

$$\begin{aligned} \sum_{e_{j_i} \in OPT \backslash S} f(e_{j_i}|S) &\leq k \cdot \frac{O}{2k} \\ &\leq O/2. \end{aligned}$$

On the other hand, we also have that

$$\begin{aligned} \sum_{e_{j_i} \in OPT \backslash S} f(e_{j_i}|S) &\geq \sum_{i=1}^{t} f(e_{j_i}|S \cup \{e_{j_1}, \dots, e_{j_{i-1}}\}) \\ &= f(S \cup \{e_{j_1}, \dots, e_{j_t}\}) - f(S) \\ &= f(OPT) - f(S), \end{aligned}$$

where in the above we use that $f(e_{j_i}|S) \geq f(e_{j_i}|S \cup \{e_{j_1}, \dots, e_{j_{i-1}}\})$ by the submodularity of f, and then let the sum of marginal gains telescope. From the two inequalities above we get that

$$f(S) \geq f(OPT) - O/2.$$

The desired lower bound now follows by simply taking the min over the two cases. □

We see that if $O = f(OPT)$, then by setting the threshold according to this value we immediately get a $1/2$-approximation algorithm. In the setting where we do not have this information, but have the promise that the optimum value lies in the range $[E, m]$, we can run multiple copies of this algorithm with a set of geometrically scaling guesses

$$\mathcal{O} = \{E, (1+\theta)E, (1+\theta)^2 E, \dots, (1+\theta)^{\lfloor \log_{1+\theta} \frac{m}{E} \rfloor} E, m\}.$$

Essentially, we try geometrically varying guesses for $f(OPT)$ so that we are assured that there is some $O^\dagger \in \mathcal{O}$ such that

$$f(OPT) \leq O^\dagger \leq (1+\theta) f(OPT).$$

From the guarantee of Theorem 1.3, we get that if $S^{O^\dagger}$ is the output of Algorithm 1 when run with $O = O^\dagger$, then

$$\begin{aligned} f(S^{O^\dagger}) &\geq \min\{O^\dagger/2, f(OPT) - O^\dagger/2\} \\ &\geq \min\{f(OPT)/2, (1-\theta) f(OPT)/2\} \\ &\geq (1-\theta) f(OPT)/2. \end{aligned}$$

We see that since we must maintain and update all the S^O for $O \in \mathcal{O}$, achieving this guarantee requires that we pay a computational and spatial overhead of $\frac{2 \log m/E}{\theta}$ for a $\frac{1-\theta}{2}$-approximation. We also note that this algorithm requires just one pass over the data stream.

B. Proof of Theorem 1.4

For ease of reference we reproduce the pseudo-code of Algorithm 2 and Algorithm 3.

Algorithm 2 Sparse, Dwork et al. (2009); Dwork & Roth (2014)

```
input  Arbitrary (possibly adaptive) stream of sensitivity 1 queries $f_1, f_2, \dots$, a threshold $T$, a cutoff point $k$, threshold
  noise $\mathcal{D}_\alpha$, score noise $\mathcal{D}_\beta$. Output is a stream of answers $a_1, a_2, \dots, a_i \in \{\bot, \top\}$
  $\alpha \sim \mathcal{D}_\alpha$
  Let $\hat{T}_i = T + \alpha$ for $i \in \{0, \dots, k-1\}$
  Let count $= 0$
  for query $e_i$ do
    Let $\beta_i \sim \mathcal{D}_\beta$
    if $f_i + \beta_i \geq \hat{T}_{\text{count}}$ then
      Output $a_i = \top$
      count $=$ count $+ 1$
    else
      $a_i = \bot$
    end if
    if count $\geq k$ then
      Halt
    end if
  end for
```

Lemma B.16. *If $\alpha \in (a_l, a_u)$ and for all elements $e_i \in V$, $\beta_i \in (b_l, b_u)$, then Algorithm 2 has the promise that when run with threshold $T = O/2k$, if we add all elements e_i for which $a_i = \top$ to S^O, then*

$$f(S^O) \geq \frac{1}{2} \min\{O, f(\mathrm{OPT}) - O\} - kb_u + ka_l.$$

Proof. Let $S^O = \{e_{i_1}, \dots, e_{i_k}\}$. Let $S^O_{i_u} := \{e_{i_1}, \dots, e_{i_{u-1}}\}$ for $u \leq k$. Since the selected elements must have succeeded in the threshold check, it must be the case that

$$\begin{aligned} f(e_{i_u} | S^O_{i_u}) + \beta_{i_u} &\geq \frac{O}{2k} + \alpha_{i_u} \\ \Rightarrow f(e_{i_u} | S^O_{i_u}) + b_u &\geq \frac{O}{2k} + a_l \\ \Rightarrow f(e_{i_u} | S^O_{i_u}) &\geq \frac{O}{2k} - b_u + a_l \end{aligned}$$

Hence we have that

$$\begin{aligned} f(S) &= \sum_{u=0}^{k} f(e_{i_u} | S^O_{i_u}) \\ &\geq \frac{O}{2} - kb_u + ka_l \end{aligned}$$

On the other hand if $|S^O| = r < k$, as in the proof of Theorem 1.3, let $\mathrm{OPT} \setminus S = \{e_{j_1}, \dots, e_{j_t}\}$, and as before, we have that

$$\begin{aligned} f(S^O) &\geq f(\mathrm{OPT}) - \sum_{e_{j_i} \in \mathrm{OPT} \setminus S^O} f(e_{j_i} | S^O) \\ &\geq f(\mathrm{OPT}) - \sum_{i=1}^{t} (O/2k + \alpha_{j_i} - \beta_{j_i}) \\ &\geq f(\mathrm{OPT}) - \frac{O}{2} - kb_u + ka_l. \end{aligned}$$

□

Algorithm 3 Private streaming submodular maximization

```
input  Monotone submodular function $f_A$, cardinality constraint parameter $k$, failure probability $\beta$, privacy parameters
   $(\varepsilon, \delta)$, element stream $V$, approximation parameter $\theta$
  if $f_A$ has sensitivity 1 but is not 1-decomposable then
    Let $\varepsilon' = \frac{\varepsilon}{4\sqrt{2T \ln((T+1)/\delta)}}$
    Let $\sigma = \frac{\sqrt{32k \ln \frac{T+1}{\delta}}}{\varepsilon'}$
    Let $\mathcal{D}_\alpha = \mathrm{Lap}(\sigma)$
    Let $\mathcal{D}_\beta = \mathrm{Lap}(2\sigma)$
  else if $f_A$ is 1-decomposable then
    $\gamma = O\left(\frac{\varepsilon}{\sqrt{T} \log^{1.5} T/\delta}\right)$ (exact expression in Lemma 4.2)
    $\mathcal{D}_\alpha = \mathrm{Gumb}(\gamma)$
    $\mathcal{D}_\beta = \mathrm{Gumb}(\gamma)$
  end if
  Let $E = \min\left\{\frac{k \log n}{\varepsilon}, m/2\right\}$
  Let $T = \lceil \log_{1+\theta} \frac{m}{E} \rceil + 1$
  Let $\mathcal{O} = \{E, (1+\theta)E, (1+\theta)^2 E, \ldots, (1+\theta)^{\lfloor \log_{1+\theta} \frac{m}{E} \rfloor} E, m\}$
  for $e_i \in V$ do
    for all $O \in \mathcal{O}$ do
      $S^O \leftarrow$ Algorithm 2
      (Query stream $(f_A(e|S^O))_{e \in V}$, threshold $\frac{O}{2k}$,
      cutoff $k, \mathcal{D}_\alpha, \mathcal{D}_\beta$)
      where

          $a_i = \top \Rightarrow$ add element $e_i$ to $S^O$
          $a_i = \bot \Rightarrow$ reject element $e_i$

    end for
  end for
  $S^{O^*} \leftarrow EM(\{S^O : O \in \mathcal{O}\}, \varepsilon/2)$
output  $S^{O^*}$
```

Remark B.17. Note that although the noise random variables β_{e_i} for $e_i \in O$ are all drawn independently and have mean 0, we have implicitly conditioned on S^O of elements having already been picked, and so we cannot claim and exploit independence so as to derive a better concentration bound. One would expect to see noise values biased high, making it more likely that that element have passed the check. Although we should be able to derive a concentration bound for the threshold noise random variables α_O that scales as $\tilde{O}(\sqrt{k})$, as the other noise term dominates in magnitude this does not help asymptotically.

Proof of Theorem 1.4. We assume for now that f has sensitivity 1. From Theorem 2.8 and the choice of σ in Algorithm 2 we see that each one of the $T = |\mathcal{O}|$ calls to is $(\varepsilon', \frac{\delta}{T+1})$-DP. Then, by advanced composition (Theorem 2.4), it follows that since ε' as set in the pseudocode is

$$\frac{\varepsilon}{4\sqrt{2T \ln(T+1)/\delta}},$$

the T calls to Algorithm 2 are collectively $\left(\varepsilon/2, T\frac{\delta}{T+1} + \frac{\delta}{T+1}\right)$-differentially private. We apply basic composition to account for the $\varepsilon/2$-private call to the exponential mechanism and get (ε, δ)-differential privacy in sum.

We now derive the the utility guarantee. For all $O \in \mathcal{O}$, and all α_i, with probability $1 - \frac{\eta}{2kT}$

$$|\alpha_i| \leq \frac{8 \log 2T/\eta}{\sigma}.$$

Similarly, with probability $1 - \frac{\eta}{2nT}$, we have that for an element e_j,

$$|\beta_j| \leq \frac{4 \log 2nT/\eta}{\sigma}.$$

Applying the union bound, it follows that with probability $1 - \eta$, for all n elements across all T thresholds as well as for all T guesses the respective noise variables α_i, β_j are bounded as above. It follows that we can apply Lemma B.16 with $a_u = -a_l = \frac{8 \log T/\eta}{\sigma}$, and $b_u = -b_l = \frac{4 \log n/\eta}{\sigma}$. Substituting, we get that for all guesses O with probability $1 - \eta/2$,

$$f(S^O) \geq \frac{1}{2} \min\left\{O, 2f(\mathrm{OPT}) - O\right\} - \frac{4k \log 2nT/\eta}{\sigma} - \frac{8k \log 2kT/\eta}{\sigma}$$

As O varies geometrically between E and m, and we have the promise that f takes values in $[E, m]$, it follows that there is a choice $O^\dagger$ such that

$$\begin{aligned} f(\mathrm{OPT}) \leq O^\dagger &\leq (1+\theta) f(\mathrm{OPT}) \\ \Rightarrow \min\{O^\dagger, 2f(\mathrm{OPT}) - O^\dagger\} &\geq (1-\theta) f(\mathrm{OPT}) \\ \Rightarrow f(S^{O^\dagger}) &\geq \frac{(1-\theta) f(\mathrm{OPT})}{2} - \frac{12k \log 2nkT/\eta}{\sigma}. \end{aligned}$$

From the guarantee of the exponential mechanism we get that with probability $1 - \eta/2$,

$$f(S^{O^*}) \geq f(S^{O^\dagger}) - \frac{2}{\varepsilon} \ln \frac{2T}{\eta}.$$

In sum, putting everything together and applying the union bound, we have that with probability $1 - \eta$,

$$\begin{aligned} f(S^{O^*}) &\geq \frac{(1-\theta) f(\mathrm{OPT})}{2} - \frac{12k \log(2nkT/\eta) \sqrt{32k \ln \frac{1}{\delta}}}{\varepsilon'} - \frac{2}{\varepsilon} \ln \frac{2T}{\eta} \\ &\geq \frac{(1-\theta) f(\mathrm{OPT})}{2} - O\left(\frac{k \sqrt{kT \log(T/\delta) \log(1/\delta)} \log nkT/\eta}{\varepsilon} \right) \\ &\geq \frac{(1-\theta) f(\mathrm{OPT})}{2} - O\left(\frac{k^{1.5} \log^{1.5} \frac{nk \log m/E}{\eta\theta\delta} \log^{0.5} m/E}{\varepsilon\sqrt{\theta}} \right). \end{aligned}$$

wherein we use that $T = O(\log_{1+\theta} m/E) = O\left(\frac{\log m/E}{\theta}\right)$.

We now set $E = \min\left\{\frac{k \log n}{\varepsilon}, m/2\right\}$, and show that the claimed bound holds. If $f(\mathrm{OPT})$ lies in the prescribed interval $[E, m]$, then we have already shown that the claimed bound holds. On the other hand, if $f(\mathrm{OPT}) < E$, the additive loss in utility is

$$\begin{aligned} \frac{k^{1.5} \log^{1.5} \frac{nk \log 2}{\eta\theta\delta} \log^{0.5} 2}{\varepsilon\sqrt{\theta}} &> \frac{k \log n}{\varepsilon} \\ &> f(\mathrm{OPT}) \end{aligned}$$

and so the RHS of the claimed bound is negative, in which case any choice of S^{O^*} fulfills the claimed bound trivially. We can therefore say that unconditionally, with probability $1 - \eta$,

$$f(S^{O^*}) \geq \frac{(1-\theta) f(\mathrm{OPT})}{2} - O\left(\frac{k^{1.5} \log^{1.5} \frac{nk}{\eta\theta\delta} \log \frac{m}{E}}{\varepsilon\sqrt{\theta}} \right).$$

In the more general case where f has sensitivity λ, we observe that our analysis holds for the function f/λ, and that the optimum utility maximizing set for f/λ is the same as that of f and its utility is $\frac{f(\mathrm{OPT})}{\lambda}$, so we have that with probability $1 - \eta$,

$$\frac{f(S^{O^*})}{\lambda} \geq \frac{(1-\theta)f(\mathrm{OPT})}{2\lambda} - O\left(\frac{k^{1.5}\log^{1.5}\frac{nk}{\eta\theta\delta}\log\frac{m}{E}}{\varepsilon\sqrt{\theta}}\right).$$

Multiplying throughout by λ leads to the stated bound for the general case. □

C. Proof of Theorem 1.5

The key technical lemma in the privacy analysis of Algorithm 3 with Gumbel noise is the following.

Lemma C.1. *Algorithm 2 is (ε, δ)-differentially private for $\mathcal{D}_\alpha, \mathcal{D}_\beta = Gumb(\gamma)$, where $\gamma = \frac{8}{\varepsilon \ln 2}\log\frac{2}{\varepsilon\delta}$.*

The proof of this result is technically involved, and we defer the proof to the end of this section. In addition to Lemma C.1, we will also need some standard concentration bounds for the Gumbel distribution.

Lemma C.2. *If $\alpha_i \sim Gumbel(\mu, \gamma)$, the following statements hold.*

1. *With probability $1-\beta$, $x \leq \mu + \gamma\log 1/\beta$.*
2. *With probability $1-\beta$, $x \geq \mu - \gamma\log\log\frac{1}{\beta}$.*

Proof. 1. We recall that the CDF for $Gumbel(\mu, \gamma)$ is $\exp(-\exp(-(x-\mu)/\gamma))$. Then,

$$\begin{aligned} 1 - \exp(-\exp(-(x-\mu)/\gamma)) &\leq \beta \\ \Leftrightarrow -\exp(-(x-\mu)/\gamma) &\geq \log 1 - \beta \\ \Leftrightarrow \frac{x-\mu}{\gamma} &\geq \log\frac{1}{\log\frac{1}{1-\beta}} \\ \Leftrightarrow x &\geq \mu + \gamma\log\frac{1}{\log\frac{1}{1-\beta}}. \end{aligned}$$

Using the series expansion $\log 1 - x = -x - x^2/2 - \cdots \leq -x$, we have that

$$\begin{aligned} \log\frac{1}{1-\beta} &= -\log 1 - \beta \\ &\geq \beta \\ \Rightarrow \log\frac{1}{\log\frac{1}{1-\beta}} &\leq \log\frac{1}{\beta} \end{aligned}$$

So in particular, if $x \geq \mu + \gamma\log\frac{1}{\beta}$, then $P(\alpha_i \geq x) \leq \beta$.

2. Similar to the first part, we have that

$$\begin{aligned} \exp(-\exp(-(x-\mu)/\gamma)) &\leq \beta \\ \Leftrightarrow \exp(-(x-\mu)/\gamma) &\geq \log\frac{1}{\beta} \\ \Leftrightarrow \frac{(x-\mu)}{\gamma} &\leq -\log\log\frac{1}{\beta} \\ \Leftrightarrow x &\leq \mu - \gamma\log\log\frac{1}{\beta}. \end{aligned}$$

□

With these lemmas we can now derive our main result.

Proof of Theorem 1.5. We assume for now that f is 1-decomposable. From Lemma C.2, we have that for every guess O_i, with probability $1 - \eta/2kT$

$$\alpha_i \in \left[-\gamma \log\log \frac{2kT}{\eta}, \gamma \log \frac{2kT}{\eta}\right].$$

Similarly, for every element e_i and every threshold O, with probability $1 - \eta/2nT$

$$\beta_i \in \left[-\gamma \log\log \frac{2nT}{\eta}, \gamma \log \frac{2nT}{\eta}\right].$$

Applying the union bound, it follows that in the notation of Lemma B.16 we can set

$$\begin{aligned} a_l &= -\gamma \log\log \frac{2kT}{\eta} \\ a_u &= \gamma \log \frac{2kT}{\eta} \\ b_l &= -\gamma \log\log \frac{2nT}{\eta} \\ b_u &= \gamma \log \frac{2nT}{\eta}. \end{aligned}$$

and conclude that with probability $1 - \eta$, for all thresholds O,

$$\begin{aligned} f(S^O) &\geq \frac{1}{2}\min\{O, f(\mathrm{OPT}) - O\} - k\gamma \log \frac{2nT}{\eta} - k\gamma \log\log \frac{2kT}{\eta} \\ &\geq \frac{1}{2}\min\{O, f(\mathrm{OPT}) - O\} - 2k\gamma \log \frac{2nkT}{\eta} \end{aligned}$$

As in the proof of Theorem 1.4, as long as $f(\mathrm{OPT}) \in [E, m]$ there exists a choice $O^\dagger$ for which

$$f(S^{O^\dagger}) \geq \frac{(1-\theta)f(\mathrm{OPT})}{2} - 2k\gamma \log \frac{2nkT}{\eta}.$$

From the utility guarantee of the exponential mechanism (Lemma 2.5) we have that

$$\begin{aligned} f(S^*) &\geq f(S^{O^\dagger}) - \frac{2}{\varepsilon} \log \frac{2T}{\eta} \\ &\geq \frac{(1-\theta)f(\mathrm{OPT})}{2} - 2k\gamma \log \frac{2nkT}{\eta} - \frac{2}{\varepsilon}\log\frac{2T}{\eta} \\ &\geq \frac{(1-\theta)f(\mathrm{OPT})}{2} - O\left(\frac{k\sqrt{T}\log^{1.5}\frac{T\log T/\delta}{\varepsilon\delta}\log\frac{2nkT}{\eta}}{\varepsilon}\right) - \frac{2}{\varepsilon}\log\frac{2T}{\eta} \\ &\geq \frac{(1-\theta)f(\mathrm{OPT})}{2} - O\left(\frac{k\log^{0.5}\frac{m}{E}\log^{1.5}\frac{\log\frac{m}{E}\log\frac{\log m/E}{\theta\delta}}{\varepsilon\delta\theta}\log\frac{2nk\log m/E}{\eta\theta}}{\varepsilon\sqrt{\theta}}\right) \end{aligned}$$

wherein we use that $\gamma = O\left(\frac{\sqrt{T}}{\varepsilon}\log^{1.5}\frac{T\log T/\delta}{\varepsilon\delta}\right)$, and $T = \log_{1+\theta} m/E = O\left(\frac{\log m/E}{\theta}\right)$. Again, setting $E = \min\left\{\frac{k\log n}{\varepsilon}, m/2\right\}$, we get that if $f(\mathrm{OPT}) \in [E, m]$, then a good choice of $O^\dagger$ exists and the desired bound follows. On the other hand, if $f(\mathrm{OPT}) < E < \frac{k\log n}{\varepsilon}$, then since the additive error term is at least $\frac{k\log n}{\varepsilon}$, the RHS of the claimed bound is negative, and any choice of S^{O^*} fulfills the claimed bound trivially. Simplifying terms a bit we get

$$f(S^*) \geq \frac{(1-\theta)f(\mathrm{OPT})}{2} - O\left(\frac{k\log^2\frac{m}{E\varepsilon\delta\theta}\log\frac{2nk\log m/E}{\eta\theta}}{\varepsilon\sqrt{\theta}}\right).$$

In the more general case where f is λ-decomposable, we observe that our analysis holds for the 1-decomposable function f/λ, and that the optimum utility maximizing set for f/λ is the same as that of f and its utility is $\frac{f(\mathrm{OPT})}{\lambda}$. It follows that with probability $1-\eta$,

$$\frac{f(S^*)}{\lambda} \geq \frac{(1-\theta)f(\mathrm{OPT})}{2\lambda} - O\left(\frac{k\log^2\frac{m}{E\varepsilon\delta\theta}\log\frac{2nk\log m/E}{\eta\theta}}{\varepsilon\sqrt{\theta}}\right).$$

Multiplying throughout by λ leads to the stated bound.

We now derive the privacy guarantee. Since the $T \leq \frac{2\log\frac{m}{E}}{\theta}$-many instantiations of $Algorithm$ 2 are run with independent random bits, we can use advanced composition to argue that the net privacy loss suffered by releasing the sets S^O is

$$(2\varepsilon'\sqrt{2T\log(1/\delta')}, (T+1)\delta'),$$

where it suffices to set $\gamma = \frac{8}{\varepsilon'\ln 2}\log\frac{2}{\varepsilon'\delta'}$ by Lemma C.1. Replacing ε' by $\frac{\varepsilon}{4\sqrt{2T\log(1/\delta')}}$ in the expression for γ, it follows that for

$$\begin{aligned}\gamma &= \frac{32\sqrt{2T\log T/\delta}}{\varepsilon\ln 2}\log\frac{4T\sqrt{2T\log(T/\delta)}}{\varepsilon\delta}\\ &= O\left(\frac{1}{\varepsilon\sqrt{\theta}}\log^{1.5}\frac{\log\frac{1}{\theta\delta}}{\theta\varepsilon\delta}\right),\end{aligned}$$

Algorithm 3 with Gumbel noise is $(\varepsilon/2, \delta)$-differentially private. We can then apply the exponential mechanism on this public set of choices for an additional $(\varepsilon/2, 0)$-privacy loss, giving us the claimed expression. □

To prove Lemma C.1, we first derive some technical lemmas that characterize the probability of stream elements succeeding in their privatized threshold checks. Lemma 4.1 characterizes the probability of an element e_i being picked condition on the set S already having been picked.

Lemma C.3. *Conditioned on having picked the set S_i of elements with $|S_i| = u < k$,*

$$\Pr[a_i = \top|S_i] = \int_{-\infty}^{\infty} 1 - \exp\Big(-w_A(i|S_i)e^{\alpha_u/\gamma}\Big)\mathrm{d}\alpha_{|S_i|},$$

where

$$w_A(i|S_i) = \exp\left(\frac{-1}{\gamma}\left(\frac{O}{2k} - f_A(e_i|S_i)\right)\right),$$

and $\alpha_u \sim Gumbel(0,\gamma)$.

Proof. We recall that for an element e_i to be picked by Algorithm 3 (which happens if and only if the output a_i of Algorithm 2 on input e_i equals $\top$), it must be the case that if S_i is the set of elements picked thus far then $|S_i| = u < k$, and that the privatized marginal utility of e_i given S_i has been picked beats the privatized threshold $O/2k + \alpha_u$. We can write

$$\begin{aligned}\Pr[a_i = \top|S_i] &= \Pr\left[f(e_i|S_i) + \beta_i \geq \frac{O}{2k} + \alpha_{|S|}\right]\\ &= \int_{-\infty}^{\infty} 1\left[f(e_i|S_i) + \beta_i \geq \frac{O}{2k} + \alpha_u\right]\mathrm{d}\alpha_u \mathrm{d}\beta_i\end{aligned}$$

where $\alpha_u, \beta_i \sim Gumbel(0,\gamma)$, and $1[\cdot]$ denotes the indicator of the event in its argument. Since α_u and β_i are drawn independently of each other, we can factorize their joint density function and write

$$\Pr[a_i = \top|S_i] = \int_{-\infty}^{\infty} \Pr\left[\beta_i \geq \frac{O}{2k} + \alpha_u - f(e_i|S_i)\right]\mathrm{d}\alpha_u. \tag{1}$$

Since $\beta_i \sim Gumbel(0, \gamma)$, we have that

$$\begin{aligned}\Pr\Big[\beta_i \geq \frac{O}{2k} + \alpha_u - f(e_i|S_i)\Big] &= 1 - \Pr\Big[\beta_i < \frac{O}{2k} + \alpha_u - f(e_i|S_i)\Big] \\ &= 1 - \exp\Big(-\exp\Big(\frac{-1}{\gamma}\Big(\frac{O}{2k} + \alpha_u - f(e_i|S_i)\Big)\Big)\Big) \\ &= 1 - \exp\Big(-w_A(i|S_i)e^{\alpha_u/\gamma}\Big).\end{aligned}$$

Substituting for the integrand in Equation (1), we get the stated result. □

Lemma C.20. *Let $\mathcal{A}(A)$ denote the set of elements indicated by Algorithm 2 to be picked for the decomposable submodular function f_A. If $S = (e_{i_1}, e_{i_2}, \ldots, e_{i_r})$, then*

$$\Pr[\mathcal{A}(A) = S] = \prod_{u=1}^{r} \frac{w_A(i_u|S_{i_u})}{(1 + \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))(1 + \sum_{j\in(i_{u-1},i_u]} w_A(j|S_{i_u}))}.$$

where $S_{i_u} = \{e_{i_1}, e_{i_2}, \ldots, e_{i_{u-1}}\}$, i.e. the set of elements already picked when element e_{i_u} is considered.

Proof. The stream $e_1, e_2, \ldots$ is given as input to Algorithm 2 and the elements $e_{i_1}, \ldots, e_{i_r}$ are picked. For this to be the case, for every $u \in \{1, \ldots, r\}$, all the elements after i_{u-1} (the element picked before i_u)), and before i_u must fail their privatized checks conditioned on $\{i_1, \ldots, i_{u-1}\}$ having already been picked, but i_u must itself succeed. In a way similar to Lemma 4.1, we can factor the joint density function of the α_u and the β_j as they are drawn independently and write

$$\begin{aligned}\Pr[\mathcal{A}(A) = S] &= \prod_{u=1}^{r} \Pr[a_{i_u} = \top|S_{i_u}] \prod_{j\in(i_{u-1},i_u)} \Pr[a_j = \bot|S_{i_u}] \\ &= \prod_{u=1}^{r} \int_{-\infty}^{\infty} (1 - \exp\Big(-w_A(i_u|u)e^{\alpha_u/\gamma}\Big)) \prod_{j\in(i_{u-1},i_u)} \exp\Big(-w_A(j|S_{i_u})e^{\alpha_u/\gamma}\Big)\mathrm{d}\alpha_u \\ &= \prod_{u=1}^{r} \int_{-\infty}^{\infty} (1 - \exp(-w_A(i_u|u)e^{z})) \exp\left(-e^{z} \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})\right) \exp\big(-z - e^{-z}\big)\mathrm{d}z\end{aligned}$$

In the above we use that the PDF of α_u is $P(x) = \frac{1}{\gamma}\exp\Big(-(\frac{x}{\gamma} + e^{-x/\gamma})\Big)$, and make the variable substitution $z = x/\gamma$. We can simplify the integrand of each factor as follows.

$$\begin{aligned}&(1 - \exp(-w_A(i_u|S_{i_u})e^{z})) \exp\left(-e^{z} \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})\right) \exp\big(-z - e^{-z}\big) \\ &= \exp\left(-z - e^{-z}(1 + \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))\right) - \exp\left(-z - e^{-z}(1 + \sum_{j\in(i_{u-1},i_u]} w_A(j|S_{i_u}))\right)\end{aligned}$$

We can integrate the summands separately; as they have the same form, to derive the resulting expression it suffices to compute the first integral, which we denote I.

$$I = \int_{-\infty}^{\infty} \exp\left(-z - e^{-z}(1 + \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))\right)\mathrm{d}z$$

Let $t = -e^{-z}(1 + \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))$. Then, $\mathrm{d}t = e^{-z}(1 + \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))\mathrm{d}z$. Substituting this variable, we get

$$\begin{aligned}I &= \frac{1}{(1 + \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))} \int_{-\infty}^{0} e^{t}\mathrm{d}t \\ &= \frac{1}{1 + \sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})}.\end{aligned}$$

It follows that

$$\begin{aligned}\Pr[\mathcal{A}(A)=S] &= \prod_{u=1}^{r} \frac{1}{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})} - \frac{1}{1+\sum_{j\in(i_{u-1},i_u]} w_A(j|S_{i_u})} \\ &= \prod_{u=1}^{r} \frac{w_A(i_u|S_{i_u})}{(1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))(1+\sum_{j\in(i_{u-1},i_u]} w_A(j|S_{i_u}))}.\end{aligned}$$

□

Proof of Lemma C.1. Let $A, B \subset \mathcal{X}$, and let $A = B \sqcup \{I\}$. Let $S = (e_{i_1}, \dots, e_{i_r})$ be any sequence of elements that can be picked by the algorithm (i.e. $r \le k$). First we show that

$$\Pr[\mathcal{A}(A)=S] \le e^{\varepsilon'} \Pr[\mathcal{A}(B)=S] + \delta$$

Substituting from Lemma C.20 and rearranging terms, we have that

$$\frac{\Pr[\mathcal{A}(A)=E]}{\Pr[\mathcal{A}(B)=E]} = \prod_{u=1}^{r} \frac{w_A(i_u|S_{i_u})}{w_B(i_u|S_{i_u})} \cdot \prod_{u=1}^{r} \frac{(1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u}))(1+\sum_{j\in(i_{u-1},i_u]} w_B(j|S_{i_u}))}{(1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))(1+\sum_{j\in(i_{u-1},i_u]} w_A(j|S_{i_u}))}.$$

We bound the two factors of this expression separately. For the first factor, we have

$$\begin{aligned}\prod_{u=1}^{r} \frac{w_A(i_u|S_{i_u})}{w_B(i_u|S_{i_u})} &= \prod_{u=1}^{r} \frac{\exp\left(\frac{-1}{\gamma}\left(\frac{O}{2k} - f_A(e_{i_u}|S_{i_u})\right)\right)}{\exp\left(\frac{-1}{\gamma}\left(\frac{O}{2k} - f_B(e_{i_u}|S_{i_u})\right)\right)} \\ &= \exp\left(\frac{-1}{\gamma}\sum_{u=1}^{r} f_B(e_{i_u}|S_{i_u}) - f_A(e_{i_u}|S_{i_u})\right) \\ &= \exp\left(\frac{1}{\gamma}\sum_{u=1}^{r} f_p(e_{i_u}|S_{i_u})\right) \\ &\le \exp(1/\gamma).\end{aligned}$$

The second factor is bounded trivially from above by 1; to see this, we observe that the following sequence of inequalities holds.

$$\begin{aligned}w_A(j|u) &= \exp\left(\frac{-1}{\gamma}\left(\frac{O}{2k} - f_A(e_i|S_{i_u})\right)\right) \\ &= \exp\left(\frac{-1}{\gamma}\left(\frac{O}{2k} - f_B(e_i|S_{i_u})\right) + \frac{f_p(e_i|S_{i_u}))}{\gamma}\right) \\ &\ge \exp\left(\frac{-1}{\gamma}\left(\frac{O}{2k} - f_B(e_i|S_{i_u})\right)\right) \\ &\ge w_B(j|u).\end{aligned}$$

In sum, it follows that any value of $\gamma \ge 1/\varepsilon$ suffices. We now show that

$$\Pr[\mathcal{A}(B)=S] \le e^{\varepsilon} \Pr[\mathcal{A}(A)=S] + \delta.$$

To this end we consider the reciprocal of the ratio we bounded for the first case, i.e.

$$\frac{\Pr[\mathcal{A}(B)=E]}{\Pr[\mathcal{A}(A)=E]} = \prod_{u=1}^{r} \frac{w_B(i_u|S_{i_u})}{w_A(i_u|S_{i_u})} \cdot \prod_{u=1}^{r} \frac{(1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}))(1+\sum_{j\in(i_{u-1},i_u]} w_A(j|S_{i_u}))}{(1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u}))(1+\sum_{j\in(i_{u-1},i_u]} w_B(j|S_{i_u}))}. \tag{2}$$

To bound this ratio, we first derive a simple relaxation.

Claim C.21. The following bound holds:

$$\frac{\Pr[\mathcal{A}(B)=E]}{\Pr[\mathcal{A}(A)=E]} \le \left(\prod_{u=1}^{r} \frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u})}\right)^2.$$

Proof. We observe that

$$\frac{w_B(i_u|S_{i_u})}{w_A(i_u|S_{i_u})} = \exp\!\left(\frac{-f_{\{p\}}(e_{i_u}|S)}{\gamma}\right) \le 1,$$

and that

$$\begin{aligned}
&\frac{w_B(i_u|S_{i_u})}{w_A(i_u|S_{i_u})} \cdot \frac{1+\sum_{j\in(i_{u-1},i_u]} w_A(j|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u]} w_B(j|S_{i_u})} \\
&= \exp\!\left(\frac{-f_{\{p\}}(e_{i_u}|S)}{\gamma}\right) \cdot \frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}) + w_A(i_u|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u}) + w_B(i_u|S_{i_u})} \\
&= \exp\!\left(\frac{-f_{\{p\}}(e_{i_u}|S)}{\gamma}\right) \cdot \frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}) + \exp\big(f_{\{p\}}(e_{i_u}|S)/\gamma\big)\cdot w_B(i_u|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u}) + w_B(i_u|S_{i_u})} \\
&\le \frac{\exp\big(-f_{\{p\}}(e_{i_u}|S)/\gamma\big) + \exp\big(-f_{\{p\}}(e_{i_u}|S)/\gamma\big)\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}) + w_B(i_u|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u}) + w_B(i_u|S_{i_u})} \\
&\le \frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u}) + w_B(i_u|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u}) + w_B(i_u|S_{i_u})} \\
&< \frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u})}.
\end{aligned}$$

The claim now follows by applying this upper bound for each factor in Equation (2) as u varies from 1 to r. □

We will now focus on bounding the expression

$$\frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u})}.$$

We first observe that this expression can be identified with the expectation of an monotonically increasing function of the marginal utility of the agent $\{p\} = A\backslash B$.

Definition C.22. Let $v_j := w_B(i_{u-1}+j|S_{i_u})$. Let P_u be a distribution over $\{\perp\}\cup\{1,\dots,i_u - i_{u-1}\}$ such that $P_u(i_u+j)\propto v_j$ and $P_u(\perp)\propto 1$. With this definition, we see that

$$\begin{aligned}
\frac{1+\sum_{j>0} w_A(i_u+j|S_{i_u})}{1+\sum_{j>0} w_B(i_u+j|S_{i_u})} &= \frac{1+\sum_{j\in(i_{u-1},i_u)} \exp\big(f_{\{p\}}(e_j|S_{i_u})/\gamma\big) w_B(i_u+j|S_{i_u})}{1+\sum_{j>0} w_B(i_u+j|S_{i_u})} \\
&= \frac{1+\sum_{j\in(i_{u-1},i_u)} \exp\big(f_{\{p\}}(e_j|S_{i_u})\big) v_j}{1+\sum_{j\in(i_{u-1},i_u)} v_j} \\
&= \mathop{\mathrm{E}}_{j\sim P_u}\left[\exp\big(f_{\{p\}}(e_j|S)/\gamma\big)\right]
\end{aligned}$$

We let $Y_u := \mathrm{E}_{j\sim P_u}[\exp\big(f_{\{p\}}(e_j|S_{i_u})/\gamma\big)]$.

At a high level the key insight is that since the sum of marginal utilities $\sum_{u=1}^{r} f_{\{p\}}(e_{i_u}|S_{i_u})$ of any agent is at most 1, the net privacy loss, which we have shown to be bounded above by a product of monotonic functions of the sequential marginal utilities may also be bounded more tightly than the $\tilde{O}(\sqrt{k}/\varepsilon)$ bound that arises from advanced composition. To prove this stronger concentration bound we first derive a moment bound on these functions of the expected marginal utility Y_u.

The complication here is that the elements picked by the algorithm affect the marginal utilities of all subsequent elements considered; we proceed by formalizing a probabilistic process capturing the behaviour of this algorithm in a manner similar to that of Gupta et al. (2010) and Chaturvedi et al. (2021).

Lemma C.23. *Consider the following k-round probabilistic process. Let $v_j := w_B(i_{u-1} + j | S_{i_u})$. In each round u, it is the case that the set of elements $S_{i_u} = \{i_1, \ldots, i_{u-1}\}$ has been picked, and the element $i_u = j + i_{u-1}$ is picked with probability*

$$p_j = \frac{1}{1 + v_1 + \cdots + v_{j-1}} \cdot \frac{v_j}{1 + v_1 + \cdots + v_j}.$$

Then, for each $q = 1, \ldots, r$, for a value of $c = \gamma/4 = \frac{2}{\varepsilon \ln 2} \log \frac{2}{\varepsilon\delta}$, the following bound holds:

$$\mathop{\mathrm{E}}_{S}\left[\prod_{u=q}^{k} Y_u^c | S_{i_q}\right] \leq 1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_q})).$$

Before we prove this lemma, we first prove a minor claim that linearizes the dependence on the moments Y_u^c on the marginal utility random variable $f_{\{p\}}(e_{i_u+j}|S_{i_u})$.

Claim C.23. For c, γ such that $1 \leq c < \gamma$,

$$Y_u^c \leq 1 + \frac{(e-1)c}{\gamma} \mathop{\mathrm{E}}_{j\sim P_u}[f_{\{p\}}(e_{i_{u-1}+j}|S_{i_u})].$$

Proof. By definition, we have that

$$Y_u^c = \left(\frac{1 + \sum_{j\in(i_{u-1},i_u)} \exp\bigl(f_{\{p\}}(e_{i_{u-1}+j}|S_{i_u})/\gamma\bigr)v_j}{1 + \sum_{j\in(i_{u-1},i_u)} v_j}\right)^c.$$

Applying Jensen's inequality, we get that

$$Y_u^c \leq \frac{1 + \sum_{j\in(i_{u-1},i_u)} \exp\bigl(cf_{\{p\}}(e_{i_{u-1}+j}|S_{i_u})/\gamma\bigr)v_j}{1 + \sum_{j\in(i_{u-1},i_u)} v_j}.$$

Since $c < \gamma$ and $f_{\{p\}}(e_{i_{u-1}+j}|S_{i_u}) \leq 1$, by applying the inequality $e^x < 1 + (e-1)x$ for $x \leq 1$, it follows that

$$\exp\bigl(cf_{\{p\}}(e_{i_{u-1}+j}|S_{i_u})/\gamma\bigr) \leq 1 + (e-1)cf_{\{p\}}(e_{i_{u-1}+j}|S_{i_u}).$$

Applying this bound and continuing, we get that

$$\begin{aligned}
Y_u^c &\leq \frac{1 + \sum_{j\in(i_{u-1},i_u)}(1 + (e-1)cf_{\{p\}}(e_{i_{u-1}+j}|S_{i_u})/\gamma)w_B(i_{u-1}+j|S_{i_u})}{1 + \sum_{j\in(i_{u-1},i_u)} w_B(i_{u-1}+j|S_{i_u})} \\
&\leq 1 + \frac{\sum_{j\in(i_{u-1},i_u)}(e-1)cf_{\{p\}}(e_{i_{u-1}+j}|S_{i_u})w_B(i_{u-1}+j|S_{i_u})/\gamma}{1 + \sum_{j\in(i_{u-1},i_u)} w_B(i_{u-1}+j|S_{i_u})} \\
&\leq 1 + \frac{(e-1)c}{\gamma} \mathop{\mathrm{E}}_{j\sim P_u}[f_{\{p\}}(e_{i_u+j}|S_{i_u})].
\end{aligned}$$

□

Proof of Lemma 4.2. We proceed by reverse induction on q. For the base case, i.e. $q = k$, we have that

$$\begin{aligned}
\mathop{\mathrm{E}}_{i_k}[Y_k^c] &\leq \mathrm{E}\left[1 + \frac{(e-1)c}{\gamma} \mathop{\mathrm{E}}_{j\sim P_k}[f_{\{p\}}(e_{i_{k-1}+j})]\right] \\
&\leq 1 + \frac{(e-1)c}{\gamma} \sup_{j>0} f_{\{p\}}(e_{i_{k-1}+j}|S_{i_k}) \\
&\leq 1 + \frac{(e-1)c}{\gamma}(1 - f_{\{p\}}(S_{i_k})) \\
&\leq 1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_k})),
\end{aligned}$$

where in the above we use that $c/\gamma = 1/4 \le \frac{1}{(e-1)\varepsilon}$ for $\varepsilon \le 1$, and that $f_{\{p\}}(e_{i_{k-1}+j}|S_{i_k}) + f_{\{p\}}(S_{i_k}) = f_{\{p\}}(S_{i_k} \cup \{e_{i_{k-1}+j}\}) \le 1$ for any choice of j. For the induction step, we assume that the statement is true for $u = q+1, \dots, k$, and derive a bound for the case $u = q$.

$$
\begin{aligned}
&\mathop{\mathrm{E}}_{i_q,\dots,i_k}\left[\prod_{u=q}^{k} Y_u^c | S_{i_q}\right] \\
&= \mathop{\mathrm{E}}_{i_q}\left[Y_q^c \cdot \mathop{\mathrm{E}}_{i_{q+1},\dots,i_k}\left[\prod_{u=q+1}^{k} Y_u^c | S_q\right] | S_{i_q}\right] \\
&\le \mathop{\mathrm{E}}_{i_q}\left[Y_q^c \left(1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_q))\right) | S_{i_q}\right] \\
&\le \mathop{\mathrm{E}}_{i_q}\left[Y_q^c \left(1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_q}) - f_{\{p\}}(e_{i_q}|S_{i_q}))\right) | S_{i_q}\right] \\
&\le \mathop{\mathrm{E}}_{i_q}\left[\left(1 + \frac{(e-1)c}{\gamma} \mathop{\mathrm{E}}_{j\sim P_q}[f_{\{p\}}(e_j|S_{i_q})]\right)\left(1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_q}) - f_{\{p\}}(e_{i_q}|S_{i_q}))\right) | S_{i_q}\right] \\
&\le 1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_q})) - \frac{1}{\varepsilon}\mathop{\mathrm{E}}_{i_q}[f_{\{p\}}(e_{i_q}|S_{i_q})|S_{i_q}] \\
&+ \frac{(e-1)}{4}\mathop{\mathrm{E}}_{i_q}\left[\mathop{\mathrm{E}}_{j\sim P_q}[f_{\{p\}}(e_j|S_{i_q})] \cdot \left(1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_q}) - f_{\{p\}}(e_{i_q}|S_{i_q}))\right)\right] \\
&\le 1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_q})) - \frac{1}{\varepsilon}\mathop{\mathrm{E}}_{i_q}[f_{\{p\}}(e_{i_q}|S_{i_q})|S_{i_q}] + \frac{(e-1)(1+1/\varepsilon)}{4}\mathop{\mathrm{E}}_{i_q}[\mathop{\mathrm{E}}_{j\sim P_q}[f_{\{p\}}(e_j|S_{i_q})]],
\end{aligned}
$$

where in the above we use that

$$
1 + \frac{1}{\varepsilon}(1 - f_{\{p\}}(S_{i_q}) - f_{\{p\}}(e_{i_q}|S_{i_q})) \le 1 + \frac{1}{\varepsilon}.
$$

It follows that it would suffice to show that the last two terms sum to at most 0. We have that

$$
\begin{aligned}
\mathop{\mathrm{E}}_{i_q}[\mathop{\mathrm{E}}_{j\sim P_q}[f_{\{p\}}(e_j|S_{i_q})]] &= \sum_{w\ge 1} p_w \mathop{\mathrm{E}}_{j\sim P_q}[f_{\{p\}}(e_j|S_{i_q})|i_q = i_{q-1} + w] \\
&= \sum_{w\ge 1} p_w \sum_{x<w} \frac{v_x}{1 + v_1 + \dots + v_{w-1}} f_{\{p\}}(e_{i_{q-1}+x}|S_{i_q})
\end{aligned}
$$

The outer expectation in the display above corresponds to i_q being picked as described by the probabilistic process (and Algorithm 3), and the expectation inside is the expression that we used to bound the privacy loss term for any one round; conditioned on the choice of i_q we recall that it is a distribution over (i_{q-1}, i_q). We switch the sums in the display above to get

$$
\begin{aligned}
\mathop{\mathrm{E}}_{i_q}[\mathop{\mathrm{E}}_{j\sim P_q}[f_{\{p\}}(e_j|S_{i_q})]] &= \sum_{x\ge 1} f_{\{p\}}(e_{i_{q-1}+x}|S_{i_q}) \sum_{w>x} \frac{v_x}{1 + v_1 + \dots + v_{w-1}} p_w \\
&\le \sum_{x\ge 1} f_{\{p\}}(e_{i_{q-1}+x}|S_{i_q}) \frac{v_x}{1 + v_1 + \dots + v_x} \sum_{w>x} p_w.
\end{aligned}
$$

Further we have

$$
\begin{aligned}
\sum_{w\ge x} p_w &= \sum_{w\ge x} \frac{1}{1 + v_1 + \dots + v_{w-1}} \cdot \frac{v_w}{1 + v_1 + \dots + v_w} \\
&= \sum_{w\ge x} \frac{1}{1 + v_1 + \dots + v_{w-1}} - \frac{v_w}{1 + v_1 + \dots + v_w} \\
&= \sum_{w\ge x} \frac{1}{1 + v_1 + \dots + v_{w-1}} - \frac{v_w}{1 + v_1 + \dots + v_w} \\
&= \frac{1}{1 + v_1 + \dots + v_{x-1}}.
\end{aligned}
$$

Substituting, we get

$$\begin{aligned}\mathop{\mathrm{E}}_{i_q}[\mathop{\mathrm{E}}_{j\sim P_q}[f_{\{p\}}(e_j|S_{i_q})]] &\leq \sum_{x\geq 1} f_{\{p\}}(e_{i_{q-1}+x}|S_{i_q})\frac{v_x}{1+v_1+\cdots+v_x}\cdot\frac{1}{1+v_1+\cdots+v_{x-1}}\\ &= \sum_{x\geq 1} p_x f_{\{p\}}(e_{i_{q-1}+x}|S_{i_q})\\ &= \mathop{\mathrm{E}}_{i_q}[f_{\{p\}}(i_q|S_{i_q})].\end{aligned}$$

So in sum, we have that

$$\begin{aligned}&\mathop{\mathrm{E}}_{i_q,\ldots,i_k}\left[\prod_{u=q}^{k} Y_u^c|S_{i_q}\right]\\ &\leq 1+\frac{1}{\varepsilon}(1-f_{\{p\}}(S_{i_q}))+\mathop{\mathrm{E}}_{i_q}[f_{\{p\}}(i_q|S_{i_q})]\left(-\frac{1}{\varepsilon}+\frac{(e-1)(1+1/\varepsilon)}{4}\right)\\ &\leq 1+\frac{1}{\varepsilon}(1-f_{\{p\}}(S_{i_q})).\end{aligned}$$

wherein we use that for $\varepsilon < 1$, $\frac{(e-1)(1+1/\varepsilon)}{4} < 1/\varepsilon$. □

Returning to the proof of Lemma C.1, we see that the probabilistic process defined and analysed in Lemma 4.2 can be identified with a run of Algorithm 2 with Gumbel noise, where the input stream has been appended with infinitely many items of 0 marginal utility - this ensures that k complete rounds are executed, but the output distribution on the non-trivial items is identical. Setting $q = 1$ in Lemma 4.2, since $f_{\{p\}}(\emptyset) = 0$, we see that

$$\begin{aligned}&\mathop{\mathrm{E}}_{S}\left[\prod_{u=1}^{k} Y_u^c|S_{i_q}\right] \leq 1+\frac{1}{\varepsilon}\\ \Rightarrow &\mathop{\mathrm{E}}_{i_1,\ldots,i_k}\left[\left(\prod_{u=1}^{k}\frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u})}\right)^c\right] \leq 1+\frac{1}{\varepsilon}\\ \Rightarrow &\mathop{\mathrm{Pr}}_{i_1,\ldots,i_k}\left[\left(\prod_{u=1}^{k}\frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u})}\right) > (1+\varepsilon)^{1/2}\right] \leq \frac{(1+1/\varepsilon)}{(1+\varepsilon)^{c/2}},\end{aligned}$$

wherein in the last step we apply Markov's inequality. Since $\varepsilon < 1$, we have that

$$\begin{aligned}\frac{(1+1/\varepsilon)}{(1+\varepsilon)^{c/2}} &\leq \frac{2/\varepsilon}{(1+\varepsilon)^{c/2}}\\ &\leq \frac{2/\varepsilon}{\exp\left(\varepsilon\ln 2\cdot\frac{c}{2}\right)},\end{aligned}$$

wherein we use that for $\varepsilon < 1$, $1+\varepsilon \geq \exp(\varepsilon\cdot\ln 2)$. Setting $c = \frac{2}{\varepsilon\ln 2}\log\frac{2}{\varepsilon\delta}$, which we note is ≥ 1, we get that

$$\begin{aligned}\frac{(1+1/\varepsilon)}{(1+\varepsilon)^{c/2}} &\leq \frac{2/\varepsilon}{\exp(\ln 2/\varepsilon\delta)}\\ &= \delta.\end{aligned}$$

It follows that with probability $1-\delta$,

$$\begin{aligned}\prod_{u=1}^{k}\frac{1+\sum_{j\in(i_{u-1},i_u)} w_A(j|S_{i_u})}{1+\sum_{j\in(i_{u-1},i_u)} w_B(j|S_{i_u})} &\leq (1+\varepsilon)^{1/2}\\ \Rightarrow \frac{\Pr[\mathcal{A}(B)=E]}{\Pr[\mathcal{A}(A)=E]} &\leq 1+\varepsilon.\end{aligned}$$

It follows that a run of Algorithm 2 with Gumbel noise with noise parameter $\gamma = 4c = \frac{8}{\varepsilon\ln 2}\log\frac{2}{\varepsilon\delta}$ is (ε,δ)-DP. □

D. Proof of Theorem 1.6

In this section we describe and prove a lower bound (Theorem 1.6) for private submodular maximization. This is a slightly weaker bound than that of Gupta et al. (2010) but is more general as it applies to the (ε, δ) instead of the $(\varepsilon, 0)$ setting. Further, it also happens to have a decomposable objective, showing that Algorithm 3 with Gumbel noise has the optimal dependence on k and ε (up to logarithmic terms).

Definition D.1 (Maximum coverage)**.** Given a set system $(U, \mathcal{S})$, i.e. a ground set U and a family $\mathcal{S}$ of subsets of U, the maximum coverage problem fixes a private target subset $R \subset U$ and a number k and asks the solver to pick $\mathcal{T} \subset \mathcal{S}$ such that $R \subset \cup_{T \in \mathcal{T}} T$ and $|\mathcal{T}| \leq k$.

We can recast this problem in the form of submodular maximization, and then construct a hard instance of maximum coverage to prove our lower bound for (ε, δ)-DP submodular maximization.

Lemma D.2 (Maximum coverage as submodular maximization)**.** *Given a set system $(U, \mathcal{S})$, and a set cover problem with a private target subset $R \subset U$ and budget k, it is easy to see that the objective*

$$|R \cap (\cup_{T \in \mathcal{T}} T)| = \sum_{e \in R} 1_{e \in T}.$$

is a decomposable submodular function with $|R|$ summands.

Theorem 1.6. *For all $0 \leq \varepsilon, \delta \leq 1$, $k \in \mathbb{N}$, $n \geq k \frac{e^{\varepsilon}-1}{\delta}$, and $c \geq \frac{4\delta}{e^{\varepsilon}-1}$, if an (ε, δ)-DP algorithm for the submodular maximization problem for decomposable objectives achieves a multiplicative approximation factor of c, it must incur additive error $\Omega((kc/\varepsilon) \log(\varepsilon/\delta))$.*

Proof. We construct a hard instance for maximum coverage. Let $(U, \mathcal{S})$ be a set system where $\mathcal{S}$ consists of all the singletons in U. Let A be a set of size k picked uniformly at random from U, and let the data set $D_A = A \times [L]$ for $L = \frac{\ln c \frac{e^{\varepsilon}-1}{\delta}}{2\varepsilon}$. Let $n := |D_A| = |A| \cdot |L|$. Let $\mathcal{T}$ be k subsets of $\mathcal{S}$ picked by the solver M . The objective we are trying to maximize is

$$f(\mathcal{T}) = \sum_{e \in D_A} 1_{\{e\} \in \mathcal{T}}.$$

Let M be any (ε, δ)-DP algorithm for the set cover problem and let $\phi = \mathrm{E}_{M,A}[(M(D_A) \cap A)/|A|]$, i.e. ϕ is the average fraction of points of A (and consequently D_A) that were recovered successfully by M. ϕ captures the average approximation factor achieved by the algorithm M over this family of hard instances.

We see that since A is of size k, and the data set D_A is simply points of A repeated with multiplicity, the collection of sets $\mathcal{T} = \{\{i\} : i \in A\}$ is a solution for this maximization problem that achieves $f(\mathrm{OPT}) = n$.

We observe that

$$\begin{aligned}
\phi &= \mathop{\mathrm{E}}_{A,M}\left[\sum_{e \in D_A} 1_{\{e\} \in \mathcal{T}}\right] / |A| \\
&= \mathop{\mathrm{E}}_{A,M} \mathop{\mathrm{E}}_{i \in A}[1_{i \in M(D_A)}] \\
&= \mathop{\mathrm{E}}_{i \in U} \mathop{\mathrm{E}}_{A,M}[1_{i \in M(D_A)} | i \in A].
\end{aligned}$$

Fixing any choice of $i \in A$, let i' be uniformly random in $U \backslash A$, and let $A' = (A \backslash \{i\}) \cup \{i'\}$; A' is hence uniformly random over $U \backslash \{i\}$. We see that there is a chain of sets $D_A^0, D_A^1, \ldots, D_A^L$ such that $D_A^0 = D_A$, $D_A^t = (D_A^{t-1} \backslash \{i\}) \cup \{i'\}$ for $t \in [L]$, and $D_A^L = D_{A'}$ (we recall that we treat data sets as multisets, allowing us to swap one copy of i for one copy of i' at a time. Since M is (ε, δ)-DP, it follows that for all $t \in [L]$,

$$\mathop{\mathrm{E}}_{M}[1_{i \in M(D_A^t)}] \geq \exp(-2\varepsilon) \mathop{\mathrm{E}}_{M}[1_{i \in M(D_A^{t-1})}] - 2\delta.$$

It follows that

$$\begin{aligned}
\mathop{\mathrm{E}}_{M}[1_{i\in M(D'_A)}] &\geq \exp(-2\varepsilon)\mathop{\mathrm{E}}_{M}[1_{i\in M(D_A^{L-1})}] - 2\delta. \\
&\geq \exp(-4\varepsilon)\mathop{\mathrm{E}}_{M}[1_{i\in M(D_A^{L-2})}] - \exp(-2\varepsilon)\cdot 2\delta - 2\delta \\
&\geq \dots \\
&\geq \exp(-2L\varepsilon)\mathop{\mathrm{E}}_{M}[1_{i\in M(D_A^{0})}] - 2\delta\left(1+\exp(-2\varepsilon)+\exp(-4\varepsilon)+\dots\right) \\
&\geq \exp(-2L\varepsilon)\mathop{\mathrm{E}}_{M}[1_{i\in M(D_A)}] - \frac{2\delta}{1-e^{-2\varepsilon}} \\
&\geq \exp(-2L\varepsilon)\mathop{\mathrm{E}}_{M}[1_{i\in M(D_A)}] - \frac{2\delta}{e^{2\varepsilon}-1}.
\end{aligned}$$

Taking the expectation over $i \in U$ and the randomness in the choice of A, we get

$$\mathop{\mathrm{E}}_{i\in U}\mathop{\mathrm{E}}_{A,M}[1_{i\in M(D_A)}|i\notin A] \geq \phi\exp(-2L\varepsilon) - \frac{2\delta}{e^{2\varepsilon}-1}.$$

It follows by the law of total expectation that

$$\mathop{\mathrm{E}}_{i\in U}\mathop{\mathrm{E}}_{A,M}[1_{i\in M(D_A)}] \geq \phi\exp(-2L\varepsilon) - \frac{2\delta}{e^{2\varepsilon}-1}.$$

The LHS is at most k/n, so rearranging terms we get

$$\left(\frac{k}{n} + \frac{2\delta}{e^{2\varepsilon}-1}\right)\exp(\varepsilon\cdot 2L) \geq \phi.$$

It follows that for $n \geq k\frac{e^{2\varepsilon}-1}{2\delta}$, and $L \leq \frac{1}{2\varepsilon}\log c\frac{e^{2\varepsilon}-1}{8\delta}$, ϕ is at most $c/2$. Hence for all $c \geq \frac{8\delta}{e^{2\varepsilon}-1}$, either the algorithm fails to achieve the multiplicative approximation factor of c, or it incurs additive error $ckL/2 = \Omega((ck/\varepsilon)\log(\varepsilon/\delta))$. □

E. Experimental data and further results

E.1. Data for graphs of Section 5

We tabulate the data recorded in our main experiments for better scrutiny. As pointed out in the Section 5, note in particular the increases in the mean clustering cost when using Laplace noise for the synthetic data set (Tables 3 and 4) as we increase the cardinality constraint k from 100 to 125. There is a small jump in the cost of the Gumbel noise as well in Table 1, but this is attributable to the variance of the experiments, as the privatizing noise parameter used is the same regardless of the value of k.

Table 1. Comparison of the mean clustering cost and variance over 20 runs for the Taxi data set with privacy parameter $\varepsilon = 0.1$ (graphed in Figure 1)

Cardinality constraint k	Random	Laplace	Gumbel	Non-private
25	1.68 (0.73)	0.79 (0.34)	0.64 (0.24)	0.17 (0)
50	1.23 (0.55)	0.68 (0.23)	0.44 (0.12)	0.17 (0)
75	1 (0.44)	0.53 (0.21)	0.39 (0.12)	0.17 (0)
100	0.88 (0.34)	0.48 (0.14)	0.32 (0.09)	0.16 (0)
125	0.76 (0.26)	0.46 (0.2)	0.34 (0.11)	0.16 (0)

Table 2. Comparison of the mean clustering cost and variance over 20 runs for the Taxi data set with privacy parameter $\varepsilon = 1$ (graphed in Figure 2)

Cardinality constraint k	Random	Laplace	Gumbel	Non-private
25	1.68 (0.73)	0.7 (0.33)	0.62 (0.28)	0.15 (0)
50	1.23 (0.55)	0.64 (0.23)	0.4 (0.12)	0.14 (0)
75	1 (0.44)	0.52 (0.22)	0.36 (0.09)	0.14 (0)
100	0.88 (0.34)	0.49 (0.14)	0.3 (0.07)	0.14 (0)
125	0.76 (0.26)	0.44 (0.16)	0.28 (0.07)	0.14 (0)

Table 3. Comparison of the mean clustering cost and variance over 20 runs for the Synthetic data set with privacy parameter $\varepsilon = 0.1$ (graphed in Figure 4)

Cardinality constraint k	Random	Laplace	Gumbel	Non-private
25	1.57 (0.48)	0.98 (0.25)	0.83 (0.23)	0.38 (0)
50	1.09 (0.34)	0.7 (0.23)	0.61 (0.12)	0.22 (0)
75	0.92 (0.31)	0.59 (0.15)	0.55 (0.1)	0.22 (0)
100	0.85 (0.21)	0.53 (0.15)	0.49 (0.1)	0.21 (0)
125	0.8 (0.26)	0.56 (0.16)	0.48 (0.09)	0.21 (0)

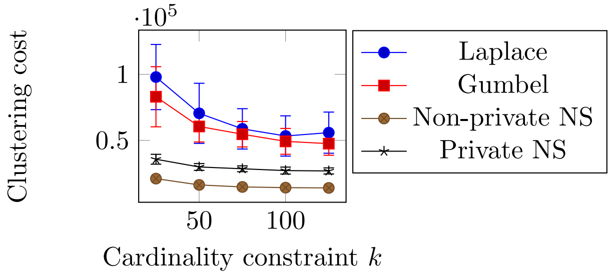
Table 4. Comparison of the mean clustering cost and variance over 20 runs for the Synthetic data set with privacy parameter $\varepsilon = 1$ (graphed in Figure 5)

Cardinality constraint k	Random	Laplace	Gumbel	Non-private
25	1.57 (0.48)	0.95 (0.27)	0.75 (0.2)	0.38 (0)
50	1.09 (0.34)	0.66 (0.2)	0.58 (0.09)	0.22 (0)
75	0.92 (0.31)	0.59 (0.14)	0.51 (0.08)	0.18 (0)
100	0.85 (0.21)	0.48 (0.11)	0.46 (0.08)	0.16 (0)
125	0.8 (0.26)	0.54 (0.16)	0.41 (0.09)	0.15 (0)

E.2. Additional experiments

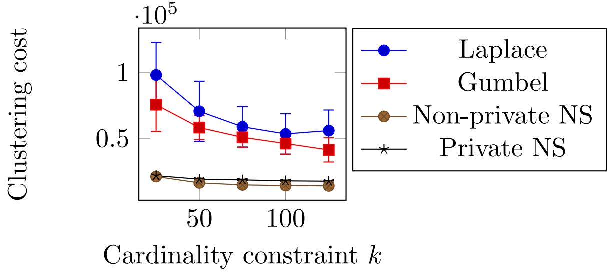
In Figure 9, we compare the performance of Algorithm 3 with **Laplace** and **Gumbel** noises with the **Private** (Mitrovic et al., 2017; Gupta et al., 2010) and the **Non-private non-streaming** (**Non-private NS**) (Nemhauser et al., 1978) algorithms. We evaluate the algorithms on the synthetic dataset described in Section 5 for $\varepsilon = 0.1$ and $\varepsilon = 1$ and $\delta = 1/|P|^{1.5}$, as before.

We see that the private non-streaming algorithm performs significantly better, as expected. We recall that the multiplicative approximation ratio for the private non streaming algorithm is $(1 - 1/e)$ which explains much of this gap; in the streaming case even for non-private algorithms the approximation factor is at best $1/2$.



(a)

Figure 7. $\varepsilon = 0.1$



(a)

Figure 8. $\varepsilon = 1$

Figure 9. Performance on the synthetic dataset.