A. Streaming Algorithms: Proofs

A.1. Proof of Theorem 4

In this section, we provide the proof of Theorem 4. Let T be the subset of $\mathcal{N}$ of size at most k maximizing $(h(r) - \varepsilon) \cdot g(T) - r \cdot \ell(T)$ among all such subsets. If $h(r) \cdot g(T) - r \cdot \ell(T) \leq 0$, then the empty set is a solution set obeying all the requirements of Theorem 4. Thus, in the rest of this section, we assume $h(r) \cdot g(T) - r \cdot \ell(T) > 0$, which implies that the value τ guessed by Algorithm 1 is positive.

The following two lemmata prove together that Algorithm 1 has the approximation guarantee of Theorem 4. The first of these lemmata handles the case in which the size of the solution S of Algorithm 1 reaches the maximum possible size k. In the proofs of both lemmata, u_i denotes the i-th element added to S by Algorithm 1.

Lemma 12. *If $|S| = k$, then $g(S) - \ell(S) \geq (h(r) - \varepsilon) \cdot g(T) - r \cdot \ell(T)$.*

Proof. Observe that

$$\begin{aligned} g(S) - \alpha(r) \cdot \ell(S) &= \sum_{i=1}^{k} [g(u_i \mid \{u_1, u_2, \ldots, u_{i-1}\}) - \alpha(r) \cdot \ell(\{u\})] \geq k\tau \\ &\geq \frac{h(r) \cdot g(T) - r \cdot \ell(T)}{1 + \varepsilon} \geq (1 - \varepsilon) \cdot h(r) \cdot g(T) - r \cdot \ell(T) \ , \end{aligned}$$

where the first inequality holds since Algorithm 1 chose to add u_i to S, and the set S at that time was equal to $\{u_1, u_2, \ldots, u_{i-1}\}$.

We now make two observations. First, we observe that

$$\alpha(r) = \frac{2r + 1 + \sqrt{4r^2 + 1}}{2} \geq \frac{1 + \sqrt{1}}{2} = 1 \ ,$$

and second, we observe that $h(r) \leq 1/2$ because

$$h(r) \leq 1/2 \iff \frac{2r + 1 - \sqrt{4r^2 + 1}}{2} \leq 1/2 \iff 2r \leq \sqrt{4r^2 + 1} \iff 4r^2 \leq 4r^2 + 1 \ .$$

Using, these observations and the above inequality, we now get

$$g(S) - \ell(S) \geq g(S) - \alpha(r) \cdot \ell(S) \geq (1 - \varepsilon) \cdot h(r) \cdot g(T) - r \cdot \ell(T) \geq (h(r) - \varepsilon) \cdot g(T) - r \cdot \ell(T) \ . \qquad \square$$

The following lemma proves the approximation ratio of Algorithm 1 for the case in which the solution set S does not reach its maximum allowed size k before the stream ends.

Lemma 13. *If $|S| < k$, then $g(S) - \ell(S) \geq (h(r) - \varepsilon) \cdot g(T) - r \cdot \ell(T)$.*

Proof. Consider an arbitrary element $u \in OPT \setminus S$. Since $|S| < k$, the fact that u was not added to S implies

$$g(u \mid S') - \alpha(r) \cdot \ell(\{u\}) < \tau \ ,$$

where S' is the set S at the time in which u arrived. By the submodularity of g, we also get

$$g(u \mid S) - \alpha(r) \cdot \ell(\{u\}) < \tau \ .$$

Adding the last inequality over all elements $u \in T \setminus S$ implies

$$\begin{aligned} g(T) - g(S) - \alpha(r) \cdot \ell(T) \leq g(T \mid S) - \alpha(r) \cdot \ell(T) &\leq \sum_{u \in T \setminus S} [g(u \mid S) - \alpha(r) \cdot \ell(\{u\})] \\ &< k\tau \leq h(r) \cdot g(T) - r \cdot \ell(T) \ , \end{aligned}$$

where the first inequality follows from the monotonicity of g, and the second inequality holds due to the submodularity of g and the non-negativity of ℓ. Rearranging this inequality yields

$$(1 - h(r)) \cdot g(T) + (r - \alpha(r)) \cdot \ell(T) < g(S) \ . \tag{2}$$

Recall that $\tau > 0$. Thus, using the same argument used in the proof of Lemma 12, we get

$$g(S) - \alpha(r) \cdot \ell(S) = \sum_{i=1}^{|S|} [g(u_i \mid \{u_1, u_2, \ldots, u_{i-1}\}) - \alpha(r) \cdot \ell(\{u\})] \geq |S|\tau \geq 0 \ .$$

Adding a $1/\alpha(r)$ fraction of this equation to a $1 - 1/\alpha(r)$ fraction of Equation (2) yields

$$g(S) - \ell(S) > (1 - 1/\alpha(r))(1 - h(r)) \cdot g(T) + (1 - 1/\alpha(r))(r - \alpha(r)) \cdot \ell(T) \ .$$

The following two calculations now complete the proof of the lemma (since $\varepsilon \cdot g(T)$ is non-negative).

$$\begin{aligned}
(1 - 1/\alpha(r))(1 - h(r)) &= \left(1 - \frac{2}{2r + 1 + \sqrt{4r^2 + 1}}\right)\left(1 - \frac{2r + 1 - \sqrt{4r^2 + 1}}{2}\right) \\
&= \frac{2r - 1 + \sqrt{4r^2 + 1}}{2r + 1 + \sqrt{4r^2 + 1}} \cdot \frac{1 - 2r + \sqrt{4r^2 + 1}}{2} = \frac{4r^2 + 1 - (2r - 1)^2}{2(2r + 1 + \sqrt{4r^2 + 1})} \\
&= \frac{(2r + 1)^2 - (4r^2 + 1)}{2(2r + 1 + \sqrt{4r^2 + 1})} = \frac{2r + 1 + \sqrt{4r^2 + 1}}{2r + 1 + \sqrt{4r^2 + 1}} \cdot \frac{2r + 1 - \sqrt{4r^2 + 1}}{2} = h(r) \ ,
\end{aligned}$$

and

$$\begin{aligned}
(1 - 1/\alpha(r))(r - \alpha(r)) &= \left(1 - \frac{2}{2r + 1 + \sqrt{4r^2 + 1}}\right)\left(r - \frac{2r + 1 + \sqrt{4r^2 + 1}}{2}\right) \\
&= -\frac{2r - 1 + \sqrt{4r^2 + 1}}{2r + 1 + \sqrt{4r^2 + 1}} \cdot \frac{1 + \sqrt{4r^2 + 1}}{2} = -\frac{2r - 1 + (4r^2 + 1) + 2r \cdot \sqrt{4r^2 + 1}}{2[2r + 1 + \sqrt{4r^2 + 1}]} \\
&= -\frac{2r \cdot [1 + 2r + \sqrt{4r^2 + 1}]}{2[2r + 1 + \sqrt{4r^2 + 1}]} = -r \ .
\end{aligned}$$

□

A.2. Proof of Corollary 6

In this section, we first restate Corollary 6 and then provide its proof.

Corollary 6. *Assume the value of β_{OPT} is given, where OPT is the optimal solution of Problem (1). Setting $r = r_{OPT} = \frac{\beta_{OPT}}{2\sqrt{1+2\beta_{OPT}}}$ makes Algorithm 1 return a solution S with the guarantee*

$$g(S) - \ell(S) \geq \left(\frac{1 + \beta_{OPT} - \sqrt{1 + 2\beta_{OPT}}}{2\beta_{OPT}} - \varepsilon'\right) \cdot (g(OPT) - \ell(OPT)) \ ,$$

where $\varepsilon' = \varepsilon \cdot (1 + 1/\beta_{OPT})$.

Proof. First, let us define $T_r^* = \arg\max_{T \in \mathcal{N}, |T| \leq k} [(h(r) - \varepsilon) \cdot g(T) - r \cdot \ell(T)]$. From Theorem 4 and the definition of T_r^*, we have

$$g(S) - \ell(S) \geq (h(r) - \varepsilon) \cdot g(T_r^*) - r \cdot \ell(T^*) \geq (h(r) - \varepsilon) \cdot g(OPT) - r \cdot \ell(OPT) \ .$$

Furthermore, from the definition of β_{OPT}, we have

$$\begin{aligned}
(h(r) - \varepsilon) \cdot g(OPT) - r \cdot \ell(OPT) &= \frac{(h(r) - \varepsilon) \cdot g(OPT) - r \cdot \ell(OPT)}{g(OPT) - \ell(OPT)} \cdot (g(OPT) - \ell(OPT)) \\
&= \frac{(h(r) - \varepsilon) \cdot (1 + \beta_{OPT}) - r}{\beta_{OPT}} \cdot (g(OPT) - \ell(OPT)) \ .
\end{aligned}$$

It can be verified that r_{OPT} is the value that maximizes the above expression, and plugging this value into the expression proves the corollary. □

A.3. Proof of Theorem 1

Recall that Theorem 1 analyzes the approximation ratio of DISTORTED-STREAMING (Algorithm 2). In Section 3, we have defined $\beta_{OPT} = \frac{g(OPT)-\ell(OPT)}{\ell(OPT)}$. In this section, we restate and prove Theorem 1. However, before doing so, let us give some intuition for DISTORTED-STREAMING. Let

$$\zeta_{OPT} = \frac{1+\beta_{OPT}-\sqrt{1+2\beta_{OPT}}}{2\beta_{OPT}} \tag{3}$$

be the approximation ratio that can be obtained for the unknown value of β_{OPT} via Corollary 6 (except for the ε' error term). One can verify that this formula for ζ_{OPT} is equivalent to the formula given for ζ_{OPT} in the statement of Theorem 1 in Section 1. The last formula implies that we always have $0 \leq \zeta_{OPT} < 1/2$. Thus, we can find an accurate guess for ζ_{OPT} by dividing the interval $[\varepsilon, 1/2)$ into small intervals (values of ζ_{OPT} below $\varepsilon < \varepsilon'$ are not of interest because Corollary 6 gives a trivial guarantee for them). Moreover, given a guess for ζ_{OPT}, we can calculate the corresponding values of β_{OPT} and r_{OPT}. DISTORTED-STREAMING is an implementation of this idea.

Theorem 1. *Despite not assuming access to β_{OPT},* DISTORTED-STREAMING *(Algorithm 2) outputs a set S obeying*

$$g(S) - \ell(S) \geq ((1-\delta') \cdot \zeta_{OPT} - \varepsilon') \cdot (g(OPT) - \ell(OPT)) \ ,$$

where $\delta' = \delta/2$ and $\varepsilon' = \frac{\varepsilon}{2\zeta_{OPT}}$.

Proof. For values of $\zeta_{OPT} < \varepsilon$, the right-hand side of the lower bounded provided by the lemma is negative, and thus, it gives a trivial lower bound (note that $\varepsilon' \geq \varepsilon$ and $\delta' > 0$ since ζ_{OPT} is always smaller than $1/2$). For this reason, in the rest of proof, we assume $\zeta_{OPT} \geq \varepsilon$. First, note that there must be a value $\zeta \in \Lambda$ such that $\zeta \leq \zeta_{OPT} < (1+\delta) \cdot \zeta$, and let us denote $\omega = \frac{\zeta_{OPT}}{\zeta}$. It is clear that $1 \leq \omega < 1+\delta$. Moreover, using the definition of ω we get that the value of β corresponding to ζ is $\beta = \frac{4\omega\zeta_{OPT}}{(\omega-2\zeta_{OPT})^2}$, and the value of r corresponding to this ζ is

$$\begin{aligned} r &= \frac{\beta}{2\sqrt{1+2\beta}} = \frac{4\omega\zeta_{OPT}/(\omega-2\zeta_{OPT})^2}{2\sqrt{1+8\omega\zeta_{OPT}/(\omega-2\zeta_{OPT})^2}} \\ &= \frac{4\omega\zeta_{OPT}}{2(\omega-2\zeta_{OPT}) \cdot \sqrt{(\omega-2\zeta_{OPT})^2+8\omega\zeta_{OPT}}} \\ &= \frac{4\omega\zeta_{OPT}}{2(\omega-2\zeta_{OPT}) \cdot \sqrt{(\omega+2\zeta_{OPT})^2}} = \frac{2\omega\zeta_{OPT}}{\omega^2-4\zeta_{OPT}^2} \ . \end{aligned}$$

To calculate the value of $h(r)$ corresponding to this value of r, we note that:

$$\sqrt{4r^2+1} = \sqrt{4\left(\frac{2\omega\zeta_{OPT}}{\omega^2-4\zeta_{OPT}^2}\right)^2+1} = \frac{\omega^2+4\zeta_{OPT}^2}{\omega^2-4\zeta_{OPT}^2} \ .$$

If we plug this equality into the definition of $h(r)$, we get

$$\begin{aligned} h(r) = \frac{2r+1-\sqrt{4r^2+1}}{2} &= \frac{1}{2} \cdot \left[1 + \frac{4\omega\zeta_{OPT}}{\omega^2-4\zeta_{OPT}^2} - \frac{\omega^2+4\zeta_{OPT}^2}{\omega^2-4\zeta_{OPT}^2}\right] \\ &= \frac{2\omega\zeta_{OPT}-4\zeta_{OPT}^2}{\omega^2-4\zeta_{OPT}^2} = \frac{2\zeta_{OPT}}{\omega+2\zeta_{OPT}} \ . \end{aligned}$$

We are now ready to plug the calculated value of r into Theorem 4, which yields that the output set S' of the instance of THRESHOLD-STREAMING initialized with this value of r obeys

$$\begin{aligned} g(S') - \ell(S') &\geq (h(r)-\varepsilon) \cdot g(OPT) - r \cdot \ell(OPT) \qquad (4) \\ &= \frac{(h(r)-\varepsilon) \cdot g(OPT) - r \cdot \ell(OPT)}{g(OPT)-\ell(OPT)} \cdot (g(OPT)-\ell(OPT)) \\ &= \frac{(h(r)-\varepsilon) \cdot (1+\beta_{OPT}) - r}{\beta_{OPT}} \cdot (g(OPT)-\ell(OPT)) \ , \qquad (5) \end{aligned}$$

where the last equality follows from the definition of β_{OPT}. Let us now lower bound the coefficient of $g(OPT) - \ell(OPT)$ in the rightmost hand side of the last equality. Recalling that $\beta_{OPT} = \frac{4\zeta_{OPT}}{(1-2\zeta_{OPT})^2}$, we get $\frac{1+\beta_{OPT}}{\beta_{OPT}} = \frac{1+4\zeta_{OPT}^2}{4\zeta_{OPT}}$. Thus, the above mentioned coefficient can be written as

$$\begin{aligned}
\frac{(h(r)-\varepsilon)\cdot(1+\beta_{OPT})-r}{\beta_{OPT}} &= \frac{1+\beta_{OPT}}{\beta_{OPT}}\cdot h(r) - \frac{r}{\beta_{OPT}} - \frac{(1+\beta_{OPT})\cdot\varepsilon}{\beta_{OPT}} \\
&= \frac{1+4\zeta_{OPT}^2}{4\zeta_{OPT}}\cdot\frac{2\zeta_{OPT}}{\omega+2\zeta_{OPT}} - \frac{\omega\cdot(1-2\zeta_{OPT})^2}{2\cdot(\omega^2-4\zeta_{OPT}^2)} - \frac{(1+4\zeta_{OPT}^2)\cdot\varepsilon}{4\zeta_{OPT}} \\
&= \frac{2\omega\zeta_{OPT}-\zeta_{OPT}-4\zeta_{OPT}^3}{\omega^2-4\zeta_{OPT}^2} - \frac{(1+4\zeta_{OPT}^2)\cdot\varepsilon}{4\zeta_{OPT}} \\
&= \left(1-\frac{(\omega-1)^2}{\omega^2-4\zeta_{OPT}^2}\right)\cdot\zeta_{OPT} - \frac{(1+4\zeta_{OPT}^2)\cdot\varepsilon}{4\zeta_{OPT}} \ .
\end{aligned}$$

We can observe that the coefficient of ζ_{OPT} on the rightmost side is a decreasing function of ω for $\omega \geq 4\zeta_{OPT}^2$. Together with the facts that $\zeta_{OPT} < 1/2$ and $\omega \geq 1$, this implies

$$\begin{aligned}
\frac{(h(r)-\varepsilon)\cdot(1+\beta_{OPT})-r}{\beta_{OPT}} &\geq \left(1-\frac{\delta^2}{(1+\delta)^2-4\zeta_{OPT}^2}\right)\cdot\zeta_{OPT} - \frac{\varepsilon}{2\zeta_{OPT}} \\
&\geq \left(1-\frac{\delta^2}{2\delta}\right)\cdot\zeta_{OPT} - \frac{\varepsilon}{2\zeta_{OPT}} = \left(1-\frac{\delta}{2}\right)\cdot\zeta_{OPT} - \frac{\varepsilon}{2\zeta_{OPT}} \ .
\end{aligned}$$

Plugging this inequality into Eq. (4), we get that the set S' produced by THRESHOLD-STREAMING for the above value of r has at least the value guaranteed by the lemma for the output set S of DISTORTED-STREAMING. The lemma now follows since the set S is chosen as the best set among multiple options including S'. □

B. Guessing τ in Algorithm 1

In this section, we explain how one can guess the value τ in Algorithm 1, which is a value obeying $k\tau \leq h(r)\cdot g(T) - r\cdot\ell(T) \leq (1+\varepsilon)k\tau$, at the cost of increasing the space complexity of the algorithm by a factor of $O(\varepsilon^{-1}(\log k + \log r^{-1}))$. Like in Appendix A.1, we assume that $h(r)\cdot g(T) - r\cdot\ell(T)$—and thus, also τ—is positive.

Observe that

$$\begin{aligned}
\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})] \leq h(r)\cdot g(T) - r\cdot\ell(T) &\leq \sum_{u\in T}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})] \\
&\leq k\cdot\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})] \ ,
\end{aligned}$$

where the first inequality holds since $\{u\}$ is a candidate to be T for every $u\in\mathcal{N}$, and the second inequality follows from the submodularity of g. Thus, if we knew the value of $\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})]$ from the very beginning, we could simply run in parallel an independent copy of Algorithm 1 for every value of τ that has the form $(1+\varepsilon)^i$ for some integer i and falls within the range

$$\left[k^{-1}\cdot\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})], (1+\varepsilon)\cdot\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})]\right] \ .$$

Clearly, at least one of the values we would have tried obeys $k\tau \leq h(r)\cdot g(T) - r\cdot\ell(T) \leq (1+\varepsilon)k\tau$, and the number of values we would have needed to try is upper bounded by

$$1+\log_{1+\varepsilon}\left(\frac{(1+\varepsilon)\cdot\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})}{k^{-1}\cdot\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})]}\right) = 2+\log_{1+\varepsilon}k = O(\varepsilon^{-1}\log k) \ .$$

Unfortunately, the value of $\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})]$ is not known to us in advance. To compensate for this, we make the following two observations. The first observation is that $k^{-1}\cdot\max_{u\in\mathcal{N}'}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})]$, where $\mathcal{N}'$ is the set of elements viewed so far, is a lower bound on the value of $k^{-1}\cdot\max_{u\in\mathcal{N}}[h(r)\cdot g(\{u\}) - r\cdot\ell(\{u\})]$. Following is the second observation, which shows that copies of Algorithm 1 with τ values that are much larger than this lower bound cannot accept any element of $\mathcal{N}'$, and thus, need not be maintained explicitly.

Observation 14. *If $\tau > (\alpha(r)/r) \cdot \max_{u \in \mathcal{N}'}[h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})]$, then Algorithm 1 accepts no element of $\mathcal{N}'$.*

Proof. Algorithm 1 accepts an element $u \in \mathcal{N}'$ if $g(u \mid S) - \alpha(r) \cdot \ell(\{u\}) \geq \tau$. However, the condition of the observation implies

$$\begin{aligned} g(u \mid S) - \alpha(r) \cdot \ell(\{u\}) &\leq g(\{u\}) - \alpha(r) \cdot \ell(\{u\}) = \frac{\alpha(r)}{r} \cdot [h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})] \\ &\leq \frac{\alpha(r)}{r} \cdot \max_{u \in \mathcal{N}'}[h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})] < \tau \ , \end{aligned}$$

where the first inequality follows from the submodularity of g, and the equality follows from the following calculation.

$$\alpha(r) \cdot h(r) = \frac{2r + 1 + \sqrt{4r^2+1}}{2} \cdot \frac{2r + 1 - \sqrt{4r^2+1}}{2} = \frac{(2r+1)^2 - (4r^2+1)}{4} = \frac{4r}{4} = r \ . \qquad \square$$

The above observations imply that it suffices to explicitly maintain a copy of Algorithm 1 for values of τ that are equal to $(1+\varepsilon)^i$ for some integer i and fall within the range

$$\left[k^{-1} \cdot \max_{u \in \mathcal{N}'}[h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})], \frac{\alpha(r)}{r} \cdot \max_{u \in \mathcal{N}'}[h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})]\right] \ . \tag{6}$$

In particular, we know that when the value of $\max_{u \in \mathcal{N}'}[h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})]$ increases (due to the arrival of additional elements), we can start a new copy of Algorithm 1 for the values of τ that have the form $(1+\varepsilon)^i$ for some integer i and now enter the range. By Observation 14, these instances will behave in exactly the same way as if they had been created at the very beginning of the stream. A formal description of the algorithm we obtain using this method is given as Algorithm 4. We note that the space complexity of this algorithm is larger than the space complexity of Algorithm 1 only by an $O(\varepsilon^{-1}(\log k + \log r^{-1}))$ factor because the number of values of the form $(1+\varepsilon)^i$ that can fall within the range (6) is at most

$$\begin{aligned} 1 + \log_{1+\varepsilon}\left(\frac{\frac{\alpha(r)}{r} \cdot \max_{u \in \mathcal{N}'}[h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})]}{k^{-1} \cdot \max_{u \in \mathcal{N}'}[h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})]}\right) &= 1 + \log_{1+\varepsilon}\left(\frac{k \cdot \alpha(r)}{r}\right) \\ &= 1 + \log_{1+\varepsilon}\left(\frac{k \cdot (2r + 1 + \sqrt{4r^2+1})}{2r}\right) \leq 1 + \log_{1+\varepsilon}(k + k/r) \\ &\leq 1 + \log_{1+\varepsilon} k + \log_{1+\varepsilon}(k/r) = O(\varepsilon^{-1}(\log k + \log r^{-1})) \ . \end{aligned}$$

Algorithm 4: DISTORTED-STREAMING: Guessing τ

Let $M \leftarrow -\infty$ and $I \leftarrow \varnothing$. `// M represents` $\max_{u \in \mathcal{N}'}[h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})]$ `and I is the list of copies of Algorithm 1 currently maintained.`
while *there are more elements in the stream* **do**
  Let u be the next element of the stream.
  Update $M \leftarrow \max\{M, h(r) \cdot g(\{u\}) - r \cdot \ell(\{u\})\}$.
  Let $J = \{i \in \mathbb{Z} \mid k^{-1}M \leq (1+\varepsilon)^i \leq r^{-1}M \cdot \alpha(r)\}$.
  Delete every copy of Algorithm 1 in I corresponding to a value $\tau = (1+\varepsilon)^i$ for an integer i that now falls outside the set J.
  Add to I a new copy of Algorithm 1 with $\tau = (1+\varepsilon)^i$ for every integer $i \in J$, unless such a copy already exists there.
  Pass the element u to all the copies of Algorithm 1 in I.
return *the set S maximizing $g(S) - \ell(S)$ among all the output sets of all the copies of Algorithm 1 in I.*

C. Proofs of Theorems 2 and 3

Theorem 2 guarantees the performance of DISTORTED-DISTRIBUTED (Algorithm 3). In Section C.1 we first restate Theorem 2 and then prove it. Then, in Section C.2 we consider a modified version of Algorithm 3 and use it to prove Theorem 3.

C.1. Proof of Theorem 2

Theorem 2. DISTORTED-DISTRIBUTED *(Algorithm 3) returns a set $D \subseteq \mathcal{N}$ of size at most k such that*

$$\mathbb{E}[g(D) - \ell(D)] \geq (1-\varepsilon)\big[(1-e^{-1})g(OPT) - \ell(OPT)\big] .$$

For simplicity, we assume that $1/\varepsilon$ is an integer from this point on, and let us define the submodular function $f(S) \triangleq g(S) - \ell(S)$. It is easy to see that f is a submodular function (although it is not guaranteed to be either monotone or non-negative). The Lovász extension of f is the function $\hat{f}\colon [0,1]^{\mathcal{N}} \to \mathbb{R}$ given by

$$\hat{f}(\mathbf{x}) = \mathop{\mathbb{E}}_{\theta \in \mathcal{U}(0,1)} \left[f\left(\{i : x_i \geq \theta\}\right) \right] \ ,$$

where $\mathcal{U}(0,1)$ is the uniform distribution within the range $[0,1]$ (Lovász, 1983). Note that the Lovász extension of a modular set function is the natural linear extension of the function. It was also proved in (Lovász, 1983) that the Lovász extension of a submodular function is convex. Finally, we need the following well-known properties of Lovász extensions, which follow easily from its definition.

Observation 15. *For every set $S \subseteq \mathcal{N}$, $\hat{f}(\mathbf{1}_S) = f(S)$. Additionally, $\hat{f}(c \cdot \mathbf{p}) \geq c \cdot \hat{f}(\mathbf{p})$ for every $c \in [0,1]$ and $\mathbf{p} \in [0,1]^{\mathcal{N}}$ whenever $f(\varnothing)$ is non-negative.*

Let us denote by DISTORTED-GREEDY(A) the set produced by DISTORTED-GREEDY when it is given the elements of a set $A \subseteq \mathcal{N}$ as input. Using this notation, we can now state the following lemma. We omit the simple proof of this lemma, but note that it is similar to the proof of (Barbosa et al., 2015, Lemma 2).

Lemma 16. *Let $A \subseteq \mathcal{N}$ and $B \subseteq \mathcal{N}$ be two disjoint subsets of $\mathcal{N}$. Suppose that, for each element $u \in B$, we have* DISTORTED-GREEDY$(A \cup \{u\})$ = DISTORTED-GREEDY(A). *Then,* DISTORTED-GREEDY$(A \cup B)$ = DISTORTED-GREEDY(A).

We now need some additional notation. Let S^* denote an optimal solution for Problem (1), and let $\mathcal{N}(1/m)$ represent the distribution over random subsets of $\mathcal{N}$ where each element is sampled independently with probability $1/m$. To see why this distribution is important, recall that $\mathcal{N}_{r,i}$ is the set of elements assigned to machine i in round i by the random partition, and that every element is assigned uniformly at random to one out of m machines, which implies that the distribution of $\mathcal{N}_{r,i}$ is identical to $\mathcal{N}(1/m)$ for every two integers $1 \leq i \leq m$ and $1 \leq r \leq \varepsilon^{-1}$. We now define for every integer $0 \leq r \leq \varepsilon^{-1}$ the set $C_r = \cup_{r'=1}^{r} \cup_{i=1}^{m} S_{r',i}$ and the vector $\mathbf{p}^r \in [0,1]^{\mathcal{N}}$ whose u-coordinate, for every $u \in \mathcal{N}$, is given by

$$p_u^r = \begin{cases} \Pr_{A \sim \mathcal{N}(1/m)}[u \notin C_{r-1} \text{ and } u \in \text{DISTORTED-GREEDY}(A \cup C_{r-1} \cup \{u\})] & \text{if } u \in S^* \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

The next lemma proves an important property of the above vectors.

Lemma 17. *For every element $u \in S^*$ and $0 \leq r \leq 1/\varepsilon$, $\Pr[u \in C_r] = \sum_{r'=1}^{r} p_u^{r'}$.*

Proof. Since u is assigned in round r' to a single machine uniformly at random,

$$\Pr[u \in C_{r'} \setminus C_{r'-1}] = \Pr[u \in \cup_{i=1}^{m} S_{r',i} \setminus C_{r'-1}] = \frac{1}{m} \sum_{i=1}^{m} \Pr[u \in S_{r',i} \setminus C_{r'-1} \mid u \in \mathcal{N}_{r',i}]$$

$$= \frac{1}{m} \sum_{i=1}^{m} \Pr[u \notin C_{r'-1} \text{ and } u \in \text{DISTORTED-GREEDY}(\mathcal{N}_{r',i} \cup (\cup_{r''=1}^{r'-1} \cup_{i'=1}^{m} S_{r'',i'})) \mid u \in \mathcal{N}_{r',i}]$$

$$= \frac{1}{m} \sum_{i=1}^{m} \Pr_{A \sim \mathcal{N}(1/m)}[u \notin C_{r'-1} \text{ and } u \in \text{DISTORTED-GREEDY}(A \cup C_{r'-1} \cup \{u\})] = p_u^{r'} \ ,$$

where the first equality holds since $C_{r'}$ can be obtained from $C_{r'-1}$ by adding to the last set all the elements of $\cup_{i=1}^{m} S_{r',i}$ that do not already belong to $C_{r'-1}$, and the last equality holds since the distribution of $\mathcal{N}_{r',i}$ conditioned on u belonging to this set is equal to the distribution of $A \cup \{u\}$ when A is distributed like $\mathcal{N}(1/m)$.

Since $C_1 \subseteq C_2 \subseteq \ldots \subseteq C_r$, the events $\Pr[u \in C_{r'} \setminus C_{r'-1}]$ must be disjoint for different values of r', which implies

$$\sum_{r'=1}^{r} p_u^{r'} = \sum_{r'=1}^{r} \Pr[u \in C_{r'} \setminus C_{r'-1}] = \Pr[u \in C_r] - \Pr[u \in C_0] = \Pr[u \in C_r] \ ,$$

where the last equality holds since $C_0 = \varnothing$ by definition. □

Using the last lemma, we can now prove lower bounds on the expected values of the sets $S_{r,i}$.

Lemma 18. *Let $\hat{g}$ and $\hat{\ell}$ be the Lovász extensions of the functions g and ℓ, respectively. Then, for every two integers $1 \leq r \leq \varepsilon^{-1}$ and $1 \leq i \leq m$,*

$$\mathbb{E}[f(S_{r,i})] \geq (1 - e^{-1}) \cdot \hat{g}(\mathbf{1}_{S^*} - \mathbf{p}^r) - \hat{\ell}(\mathbf{1}_{S^*} - \mathbf{p}^r) \ ,$$

and

$$\mathbb{E}[f(S_{r,i})] \geq (1 - e^{-1}) \cdot \hat{g}(\textstyle\sum_{r'=1}^{r-1} \mathbf{p}^{r'}) - \hat{\ell}(\sum_{r'=1}^{r-1} \mathbf{p}^{r'}) \ .$$

Proof. Let $R = \{u \in S^* \mid u \notin \textsc{Distorted-Greedy}(\mathcal{N}_{r,i} \cup C_{r-1} \cup \{u\})\}$, and let $O_{r,i}$ be some random subset of S^* to be specified later which includes only elements of $\mathcal{N}_{r,i} \cup C_{r-1} \cup R$. By Lemma 16,

$$\begin{aligned} S_{r,i} &= \textsc{Distorted-Greedy}(\mathcal{N}_{r,i} \cup (\cup_{r'=1}^{r-1} \cup_{i'=1}^{m} S_{r',i'})) \\ &= \textsc{Distorted-Greedy}(\mathcal{N}_{r,i} \cup C_{r-1}) = \textsc{Distorted-Greedy}(\mathcal{N}_{r,i} \cup C_{r-1} \cup R) \ . \end{aligned}$$

Due to this equality and the fact that $|O_{r,i}| \leq |S^*| \leq k$, the guarantee of Distorted-Greedy (Harshaw et al., 2019, Theorem 3) implies:

$$f(S_{r,i}) = g(S_{r,i}) - \ell(S_{r,i}) \geq (1 - e^{-1}) \cdot g(O_{r,i}) - \ell(O_{r,i}) \ .$$

Therefore,

$$\begin{aligned} \mathbb{E}[f(S_{r,i})] &\geq \mathbb{E}[(1 - e^{-1}) \cdot g(O_{r,i}) - \ell(O_{r,i})] = (1 - e^{-1}) \cdot \mathbb{E}[g(O_{r,i})] - \mathbb{E}[\ell(O_{r,i})] \\ &\geq (1 - e^{-1}) \cdot \hat{g}(\mathbb{E}[\mathbf{1}_{O_{r,i}}]) - \hat{\ell}(\mathbb{E}[\mathbf{1}_{O_{r,i}}]) \ , \end{aligned} \tag{7}$$

where the second inequality holds since $\hat{g}$ is convex and $\hat{\ell}$ is linear (see the discussion before Observation 15).

To prove the first part of the lemma, we now choose

$$O_{r,i} = (C_{r-1} \cap S^*) \cup R = (C_{r-1} \cap S^*) \cup \{u \in S^* : u \notin \textsc{Distorted-Greedy}(\mathcal{N}_{r,i} \cup C_{r-1} \cup \{u\})\} \ .$$

One can verify that this choice obeys our assumptions about $O_{r,i}$; and moreover, since the distribution of $\mathcal{N}_{r,i}$ is the same as that of $\mathcal{N}(1/m)$, we get:

$$\Pr[u \in O_{r,i}] = 1 - \Pr[u \notin O_{r,i}] = 1 - p_u^r \quad \forall\, u \in S^* \qquad \text{and} \qquad \mathbb{E}\left[\mathbf{1}_{O_{r,i}}\right] = \mathbf{1}_{S^*} - \mathbf{p}^r \ .$$

The first part of the lemma now follows by combining the last equality with Inequality (7).

To prove the second part of this lemma, we choose $O_{r,i} = C_{r-1} \cap S^*$. One can verify that this choice again obeys our assumptions about $O_{r,i}$; and moreover, by Lemma 17, $\mathbb{E}[\mathbf{1}_{O_{r,i}}] = \sum_{r'=1}^{r-1} \mathbf{p}^{r'}$. The second part of the lemma now follows by combining this equality with Inequality (7). □

We are now ready to prove Theorem 2.

Proof of Theorem 2. Let D be the output set of Algorithm 3. The definition of D and Lemma 18 together guarantee that for every $1 \leq r \leq \varepsilon^{-1} - 1$ we have

$$\mathbb{E}[f(D)] \geq \mathbb{E}[f(S_{r,1})] \geq (1 - e^{-1}) \cdot \hat{g}(\mathbf{1}_{S^*} - \mathbf{p}^r) - \hat{\ell}(\mathbf{1}_{S^*} - \mathbf{p}^r) \ ,$$

and additionally,

$$\mathbb{E}[f(D)] \geq \mathbb{E}[f(S_{1/\varepsilon,1})] \geq (1 - e^{-1}) \cdot \hat{g}(\textstyle\sum_{r=1}^{1/\varepsilon-1} \mathbf{p}^r) - \hat{\ell}(\sum_{r=1}^{1/\varepsilon-1} \mathbf{p}^r) \ .$$

Therefore,

$$\begin{aligned}\mathbb{E}[f(D)] &\geq \varepsilon \cdot \sum_{r=1}^{1/\varepsilon-1} [(1-e^{-1}) \cdot \hat{g}(\mathbf{1}_{S^*} - \mathbf{p}^r) - \hat{\ell}(\mathbf{1}_{S^*} - \mathbf{p}^r)] \\ &\qquad + \varepsilon[(1-e^{-1}) \cdot \hat{g}(\textstyle\sum_{r=1}^{1/\varepsilon-1} \mathbf{p}^r) - \hat{\ell}(\sum_{r=1}^{1/\varepsilon-1} \mathbf{p}^r)] \\ &\geq (1-e^{-1}) \cdot \hat{g}\left(\varepsilon \cdot \sum_{r=1}^{1/\varepsilon-1} (\mathbf{1}_{S^*} - \mathbf{p}^r) + \varepsilon \cdot \sum_{r=1}^{1/\varepsilon-1} \mathbf{p}^r\right) - \hat{\ell}\left(\varepsilon \cdot \sum_{r=1}^{1/\varepsilon-1} (\mathbf{1}_{S^*} - \mathbf{p}^r) + \varepsilon \cdot \sum_{r=1}^{1/\varepsilon-1} \mathbf{p}^r\right) \\ &= (1-e^{-1}) \cdot \hat{g}\left((1-\varepsilon) \cdot \mathbf{1}_{S^*}\right) - \hat{\ell}\left((1-\varepsilon) \cdot \mathbf{1}_{S^*}\right) \geq (1-\varepsilon) \cdot \left[(1-e^{-1}) \cdot g(S^*) - \ell(S^*)\right] \enspace,\end{aligned}$$

where the second inequality holds since $\hat{\ell}$ is linear and $\hat{g}$ is convex, and the last inequality follows again from the linearity of $\hat{\ell}$ and Observation 15 because $f(\varnothing) = g(\varnothing) - \ell(\varnothing) = g(\varnothing) \geq 0$. □

C.2. Proof of Theorem 3

Let us begin by restating Theorem 3.

Theorem 3. *Given a hereditary set system $(\mathcal{N}, \mathcal{I})$ of rank R (i.e., $R = \max_{S \in \mathcal{I}} |S|$). If the greedy algorithm obtains α-approximation for the problem of finding a set $S \in \mathcal{I}$ maximizing a given non-negative monotone submodular function $f \colon 2^{\mathcal{N}} \to \mathbb{R}_{\geq 0}$, then, for every $\varepsilon > 0$ and number m of machines, there exists a MapReduce algorithm for this problem that (i) uses $\tilde{O}(\varepsilon^{-1})$ MapReduce rounds, (ii) has a space complexity of $O(|\mathcal{N}|/m + mR/\varepsilon)$ per machine (with high probability) and $\tilde{O}(|\mathcal{N}| + m^2 R/\varepsilon)$ in total, and (iii) has an approximation ratio of $\alpha - \varepsilon$.*

To prove Theorem 3, we consider a modified version of Algorithm 3 that uses the standard greedy algorithm instead of DISTORTED-GREEDY. In the following, we refer to this modified version as GREEDY-DISTRIBUTED. Based on the pseudocode of the algorithm, we immediately get that GREEDY-DISTRIBUTED uses only $\lceil \varepsilon^{-1} \rceil = O(\varepsilon^{-1})$ MapReduce rounds. The next lemma analyzes the total and machine space complexities of GREEDY-DISTRIBUTED.

Lemma 19. *The total space complexity used by* GREEDY-DISTRIBUTED *is $O(|\mathcal{N}| + m^2 R/\varepsilon)$. Furthermore, each individual machine used by the algorithm uses, with high probability, a space complexity of $O(|\mathcal{N}|/m + mR\varepsilon)$.*

Proof. Except for the values of the variables r and i, which require only $\tilde{O}(1)$ space, GREEDY-DISTRIBUTED requires space only for two purposes: storing the elements of the input, and storing the solutions $S_{r,i}$ produced during its execution. Let us analyze the space required for these purposes separately.

- Since only $O(m/\varepsilon)$ solutions $S_{r,i}$ are produced by the algorithm during its execution, and each solution is of size at most R, the total space required to store these solutions is $\tilde{O}(mR/\varepsilon)$. As all the solutions have to be stored in each one of the m machines, they contribute $\tilde{O}(mR/\varepsilon)$ to the space complexity of each machine and $\tilde{O}(m^2 R/\varepsilon)$ to the total space complexity.

- The elements of the input are redistributed between the machines at the beginning of every MapReduce round, but they are never duplicated. Thus, they contribute $\tilde{O}(|\mathcal{N}|)$ to the total space complexity of the algorithm. If $m \geq |\mathcal{N}|/\varepsilon$, then this completes the proof of the lemma since $mR\varepsilon \geq |\mathcal{N}|$ in this case, which means that the lemma only guarantees that the space complexity used by the individual machines is at most $\tilde{O}(|\mathcal{N}|)$. Otherwise, each individual machines gets in every MapReduce round a subset of the input elements whose size follows the binomial distribution $B(|\mathcal{N}|, \varepsilon)$, and therefore, by the Chernoff and union bounds, the probability that in any iteration any machine gets more than $2|\mathcal{N}|/\varepsilon$ elements is at most
$$m\lceil \varepsilon^{-1} \rceil \cdot e^{-|\mathcal{N}|/3} \enspace,$$
which approaches 0 when the size of the ground set $|\mathcal{N}|$ increases given our assumption that $m < |\mathcal{N}|/\varepsilon$. □

It remains to analyze the approximation guarantee of GREEDY-DISTRIBUTED. Using an argument that is completely analogous to the one used in Section C.1 to prove Theorem 2, we can get that the value of the output set of GREEDY-DISTRIBUTED is, in expectation, at least $(1-\varepsilon) \cdot \alpha \cdot f(S^*) \geq (\alpha - \varepsilon) \cdot f(S^*)$.

D. Mode Finding of SLC Distributions: Proofs and Experiments

D.1. Proofs

In this section, we restate the theoretical results from Section 5 and then prove them.

Lemma 8. *For a γ-additively weak submodular function ρ, the function $\Lambda(S) \triangleq \rho(S) - \frac{\gamma}{2} \cdot |S| \cdot (|S| - 1)$ is submodular.*

Proof. For every set S and two distinct elements $u, v \notin S$, the γ-additively weak submodularity of ρ implies

$$\rho(S) + \rho(S \cup \{u, v\}) \leq \gamma + \rho(S \cup \{u\}) + \rho(S \cup \{v\}) \ .$$

Rearranging this inequality now gives

$$\begin{aligned}\rho(S) - \frac{\gamma \cdot |S| \cdot (|S| - 1))}{2} + \rho(S \cup \{u, v\}) - \frac{\gamma \cdot (|S| + 2) \cdot (|S| + 1)}{2} \\ \leq \rho(S \cup \{u\}) + \rho(S \cup \{v\}) - \frac{2 \cdot \gamma \cdot (|S| + 1) \cdot |S|}{2} \ ,\end{aligned}$$

which, by the definition of Λ, is equivalent to

$$\Lambda(S) + \Lambda(S \cup \{u, v\}) \leq \Lambda(S \cup \{u\}) + \Lambda(S \cup \{v\}) \ . \qquad \square$$

Lemma 9. *The function $g(S) \triangleq \Lambda(S) + \ell(S)$ is monotone and submodular. Furthermore, if $\rho(\varnothing) \geq 0$, then $g(S)$ is also non-negative because $\ell(\varnothing) = 0$.*

Proof. To see that $g(S)$ is submodular, recall that $\Lambda(S)$ is submodular and that the summation of a submodular function with a modular function is still submodular. To prove the monotonicity of $g(S)$, we show that for all sets $S \subseteq \mathcal{N}$ and elements $u \in \mathcal{N} \setminus S$: $g(u \mid S) \geq 0$.

$$\begin{aligned}g(u \mid S) = \Lambda(u \mid S) + \ell(u \mid S) = \Lambda(u \mid S) + \ell_u = \Lambda(u \mid S) + \max\{\Lambda(\mathcal{N} \setminus u) - \Lambda(\mathcal{N}), 0\} \\ \geq \Lambda(u \mid S) + \Lambda(\mathcal{N} \setminus u) - \Lambda(\mathcal{N}) = \Lambda(u \mid S) - \Lambda(u \mid \mathcal{N} \setminus u) \geq 0 \ ,\end{aligned}$$

where the last inequality follows from the submodularity of Λ. $\square$

Corollary 10. *Assume $\rho \colon 2^{\mathcal{N}} \to \mathbb{R}$ is a γ-additively weak submodular function obeying $\rho(\varnothing) \geq 0$. Then, when given Λ as the objective function,* DISTORTED-DISTRIBUTED *(Algorithm 3) returns a solution R such that*

$$\begin{aligned}\mathbb{E}[\rho(R)] \geq (1 - \varepsilon) \left[(1 - e^{-1})\rho(\texttt{OPT}) - e^{-1} \cdot \ell(\texttt{OPT})\right] \\ - \frac{\gamma \cdot [(1 - e^{-1}) \cdot l(l - 1) - \mathbb{E}[|R|(|R| - 1)]]}{2} \ ,\end{aligned}$$

where $\texttt{OPT} \in \arg\max_{|S| \leq k} \rho(S)$ and $l = |\texttt{OPT}| \leq k$.

Proof. Using the guarantee of Theorem 2 for the performance of DISTORTED-DISTRIBUTED for maximizing the function $\Lambda(S) = g(S) - \ell(S)$ in the distributed setting under a cardinality constraint k, we get

$$\mathbb{E}[g(R) - \ell(R)] \geq (1 - \varepsilon) \cdot \left[(1 - e^{-1}) \cdot g(\texttt{OPT}) - \ell(\texttt{OPT})\right] \ ,$$

which implies, by the definition of g,

$$\frac{\mathbb{E}[\Lambda(R)]}{1 - \varepsilon} \geq (1 - e^{-1}) \cdot (\Lambda(\texttt{OPT}) + \ell(\texttt{OPT})) - \ell(\texttt{OPT}) = (1 - e^{-1}) \cdot \Lambda(\texttt{OPT}) - e^{-1} \cdot \ell(\texttt{OPT}) \ .$$

Using the definition of Λ now, we finally get

$$\begin{aligned}\mathbb{E}[\rho(R)] \geq (1 - \varepsilon) \cdot \left[(1 - e^{-1}) \cdot \rho(\texttt{OPT}) - e^{-1} \cdot \ell(\texttt{OPT})\right] \\ - \frac{\gamma \cdot [(1 - \varepsilon) \cdot (1 - e^{-1}) \cdot |\texttt{OPT}| \cdot (|\texttt{OPT}| - 1) - \mathbb{E}[|R| \cdot (|R| - 1)]]}{2} \ ,\end{aligned}$$

which implies the corollary since $(1 - e^{-1}) \cdot |\texttt{OPT}| \cdot (|\texttt{OPT}| - 1)$ is non-negative. $\square$

Corollary 11. *Assume $\rho\colon 2^{\mathcal{N}} \to \mathbb{R}$ is a γ-additively weak submodular function obeying $\rho(\varnothing) \geq 0$. Then, when given Λ as the objective function,* THRESHOLD-STREAMING *(Algorithm 1) returns a solution R such that $\rho(R)$ is at least*

$$(h(r)-\varepsilon)\cdot\rho(\texttt{OPT}) - (\alpha(r)-r-1+\varepsilon)\cdot\ell(\texttt{OPT}) \\ - \frac{\gamma\cdot[(h(r)-\varepsilon)\cdot l\cdot(l-1) - |R|\cdot(|R|-1)]}{2} \ ,$$

where $\texttt{OPT} \in \arg\max_{|S|\leq k}\rho(S)$ and $l = |\texttt{OPT}| \leq k$.

Proof. By Theorem 4,

$$g(R) - \ell(R) \geq (h(r)-\varepsilon)\cdot g(\texttt{OPT}) - r\cdot\ell(\texttt{OPT}) \ ,$$

which implies, by the definition of g,

$$\Lambda(R) \geq (h(r)-\varepsilon)\cdot(\Lambda(\texttt{OPT})+\ell(\texttt{OPT})) - r\cdot\ell(\texttt{OPT}) = (h(r)-\varepsilon)\cdot\Lambda(\texttt{OPT}) - (r-h(r)+\varepsilon)\cdot\ell(\texttt{OPT}) \ .$$

Using the definition of Λ now, we finally get

$$\rho(R) \geq (h(r)-\varepsilon)\cdot\rho(\texttt{OPT}) - (r-h(r)+\varepsilon)\cdot\ell(\texttt{OPT}) \\ - \frac{\gamma\cdot[(h(r)-\varepsilon)\cdot|\texttt{OPT}|\cdot(|\texttt{OPT}|-1) - |R|\cdot(|R|-1)]}{2} \ .$$

The corollary now follows by observing that $r - h(r) = \frac{\sqrt{4r^2+1}-1}{2} = \alpha(r) - r - 1$. □

D.2. Experimental Evaluations

In this supplementary section, we compare the performance of DISTORTED-STREAMING with the performance of distorted greedy, vanilla greedy and sieve streaming on the problem of mode finding for an SLC distribution. We consider a distribution $\nu(S) \propto \sqrt{\det(L_S)}\cdot\mathbf{1}\{|S|\leq d\}$, where L is an $n\times n$ PSD matrix. Here, L_S corresponds to the submatrix of L, where the rows and columns are indexed by elements of S (Robinson et al., 2019). In the optimization procedure, our goal is to maximize $\rho(S) \triangleq \log(\nu(S))$. To generate the random matrix L, we first sample a diagonal matrix D and a random PSD matrix Q, and then assign $L = QDQ^{-1}$. Each diagonal element of D is from a log-normal distribution with a probability mass function $p(x) = \frac{1}{\sigma x\sqrt{2\pi}}\exp(-\frac{(\ln(x)-\mu)^2}{2\sigma^2})$, where μ and σ are the mean and standard deviation of the normally distributed logarithm of the variable, respectively. This log-normal distribution allows us to have a PSD matrix where the eigenvalues have a heavy-tailed distribution. In these experiments, we set $n = 1000, d = 100, \mu = 1.0$ and $\sigma = 1.0$.

In Fig. 5, we observe that the outcome of DISTORTED-STREAMING outperforms sieve streaming. This is mainly a result of the fact that DISTORTED-STREAMING estimates the value of $\beta_{\texttt{OPT}}$ and uses the best possible value for r. Furthermore, we see that vanilla greedy performs better than distorted greedy, and for cardinality constraints larger than $k = 20$, the performance of distorted greedy degrades. This observation could be explained by the fact that the linear cost for each element u is comparable to the value of $g(u)$ (or marginal gain of u to any set S). Therefore, distorted greedy does not pick any element in the first few iterations when k is large enough, i.e., when $(1-\frac{1}{k})^{k-(i+1)}$ is small. It is worth mentioning that while the performance of the greedy algorithm is the best for this specific application, only DISTORTED-STREAMING and distorted greedy have a theoretical guarantee.

E. Supplementary Material for Section 6: Additional Applications

E.1. Yelp Location Summarization

In this summarization task, we want to summarize a dataset of business locations. For this reason, we consider a subset of Yelp's businesses, reviews and user data (Yelp, b), referred to as the Yelp Academic dataset (Yelp, a). This dataset contains information about local businesses across 11 metropolitan areas. The features for each location are extracted from the description of that location and related user reviews. The extracted features cover information regarding several attributes, including parking options, WiFi access, having vegan menus, delivery options, possibility of outdoor seating, being good for groups.[8]

[8]For the feature extraction, we used the script provided at https://github.com/vc1492a/Yelp-Challenge-Dataset.

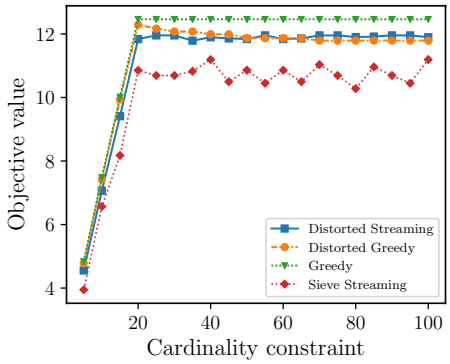


Figure 5. We want to find the mode of a distribution $\nu(S) \propto \sqrt{\det(L_S)} \cdot \mathbf{1}\{|S| \leq d\}$ for a PSD matrix L. For the objective value, we report $\log(\nu(S))$.

The goal is to choose a subset of businesses locations, out of a ground set $\mathcal{N} = \{1, \ldots, n\}$, which provides a good summary of all the existing locations. We calculated the similarity matrix $M \in \mathbb{R}^{n \times n}$ between locations using the same method described in Section 6.3. For a selected set S, we assume each location $i \in \mathcal{N}$ is represented by the location from the set S with the highest similarity to i. This makes it natural to define the total utility provided by set S using the set function

$$f(S) \triangleq g(S) - \ell(S) = \frac{1}{n} \sum_{i=1}^{n} \max_{j \in S} M_{i,j} - \sum_{u \in S} \ell_u \quad . \tag{8}$$

Note that $g(S)$ is monotone and submodular (Krause & Golovin, 2012; Frieze, 1974). For the linear function ℓ we consider two scenarios: (i) in the first one, the cost assigned to each location is defined as its distance to the downtown in the city of that location. ii) in the second scenario, the linear cost of each location u is the distance between u and the closest international airport in that area. The intuitive explanation of the first linear function is that while we look for the most diverse subset of locations as our summary, we want those locations to be also close enough to the down-town in order to make commute and access to other facilities easier. For the second linear function, we want the selected locations to be in the vicinity of airports.

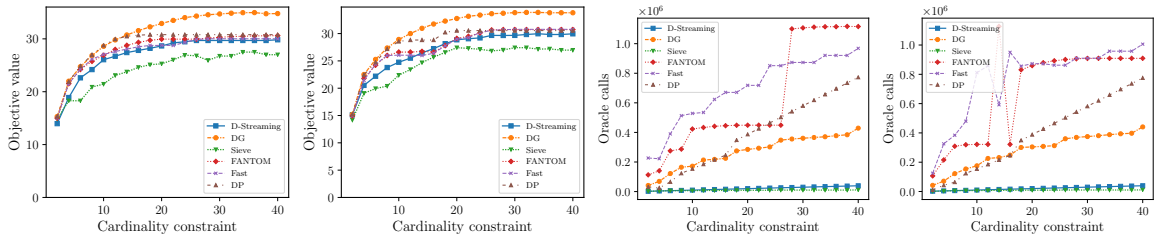
From Eq. (8) it is evident that computing the objective function requires access to the entire dataset $\mathcal{N}$, which in the streaming setting is not possible. Fortunately, however, this function is additively decomposable (Mirzasoleiman et al., 2013) over the ground set $\mathcal{N}$. Therefore, it is possible to estimate Eq. (8) arbitrarily close to its exact value as long as we can sample uniformly at random from the data stream (Badanidiyuru et al., 2014, Proposition 6.1). In this section, to sample randomly from the data stream and to have an accurate estimate of the function, we use the reservoir sampling technique explained in (Badanidiyuru et al., 2014, Algortithm 4).

In Fig. 6, we compare algorithms for varying values of k while we consider the two different linear functions ℓ. We observe that distorted greedy returns the solutions with the highest utilities. The performance of DISTORTED-STREAMING is comparable with that of the offline algorithms, and it clearly surpasses sieve-streaming. In addition, our experiments demonstrate that DISTORTED-STREAMING (and similarly sieve-streaming) requires orders of magnitude fewer oracle evaluations.

E.2. Movie Recommendation

In this application, the goal is to recommend a set of movies to a user, where we know that the user is mainly interested in movies released around 1990. As a matter of fact, we are aware that her all-time favorite movie is Goodfellas (1990). To design our recommender system, we use ratings from MovieLens users (Harper & Konstan, 2015), and apply the method of Lindgren et al. (2015) to generate a set of features for each movie.

For a ground set of movies $\mathcal{N}$, assume v_i represents the feature vector of the i-th movie. Following the same approach we used in Section 6.3, we define a similarity matrix M such that $M_{ij} = e^{-\text{dist}(v_i, v_j)}$, where $\text{dist}(v_i, v_j)$ is the euclidean distance between vectors $v_i, v_j \in \mathcal{N}$. The objective of each algorithm is to select a subset of movies that maximizes $f(S) \triangleq g(S) - \ell(S) = \log\det(\mathbf{I} + \alpha M_S) - \sum_{v \in S} \ell_v$ subject to a cardinality constraint k. In this application for

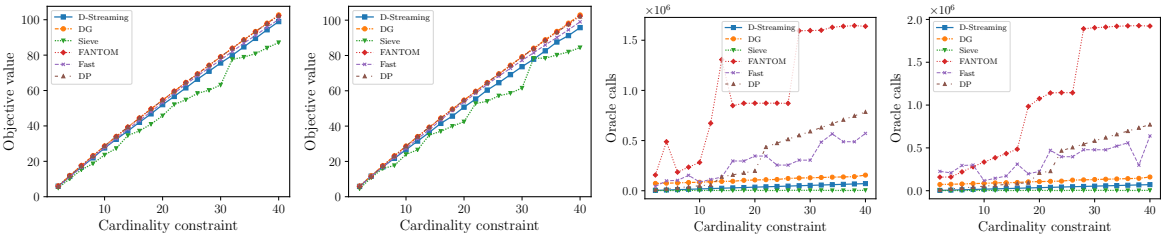


(a) ℓ = distance to down-town (b) ℓ = distance to airport (c) ℓ = distance to down-town (d) ℓ = distance to airport

Figure 6. Yelp location summarization: data points are locations from six different cities. For the linear costs we consider two different cases: 1) distances to the downtown in each city, 2) distances to the airport in each city.

$\ell(S) = \sum_{v \in S} \ell_v$ we consider two different scenarios: (i) $\ell_v = |1990 - \text{year}_v|$, where year_v denote the release year of movie v, and (2) $\ell_v = 10 - \text{rating}_v$, where rating_v denotes the IMDb rating of v (10 is the maximum possible rating).

From our experimental evaluation in Fig. 7, we observe that both modeling approaches (directly maximizing the function f and maximizing the function g subject to a knapsack constraint for ℓ) return solutions with similar objective values. Besides, we note that the computational complexity of DISTORTED-STREAMING is better than the complexity of the expensive offline algorithms (as it makes only a single pass over the data), but this difference is not very significant for some offline algorithms. Nevertheless, DISTORTED-STREAMING always provides better utility than sieve streaming.



(a) $\ell_v = |1990 - \text{year}_v|$ (b) $\ell_v = |10 - \text{rating}_v|$ (c) $\ell_v = |1990 - \text{year}_v|$ (d) $\ell_v = |10 - \text{rating}_v|$

Figure 7. Movie recommendation: we compare algorithms for varying cardinality constraint k. We use two different linear functions: $|1990 - \text{year}_v|$ and $10 - \text{rating}_v$, where the goal of first one is to recommend movies with a release year closer to 1990 and the goal of the second linear function is to promote movies with higher ratings.

E.3. Twitter Text Summarization

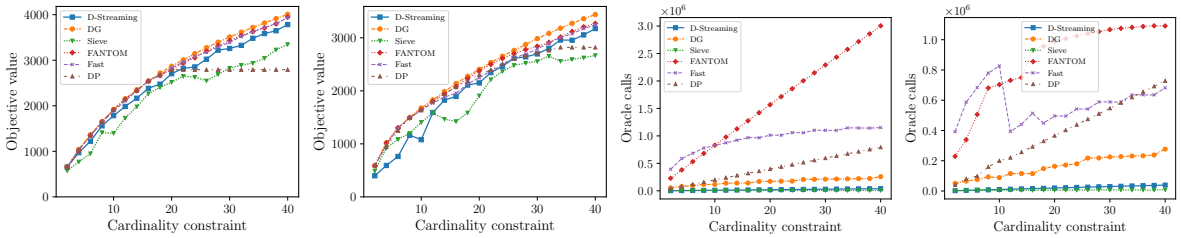
There are several news-reporting Twitter accounts with millions of followers. In this section, our goal is to produce real-time summaries for Twitter feeds of a subset of these accounts. In the Twitter stream summarization task, one might be interested in a representative and diverse summary of events that happen around a certain date. For this application, we consider the Twitter dataset provided in (Kazemi et al., 2019), where the keywords from each tweet are extracted and weighted proportionally to the number of retweets the post received. Let $\mathcal{W}$ denote the set of all existing keywords. The function f we want to maximize is defined over a ground set $\mathcal{N}$ of tweets (Kazemi et al., 2019). Assume each tweet $e \in \mathcal{N}$ consists of a positive value val_e representing the number of retweets it has received (as a measure of the popularity and importance of that tweet) and a set of l_e keywords $\mathcal{W}_e = \{w_{e,1}, \cdots, w_{e,l_e}\}$ from $\mathcal{W}$. The score of a word $w \in \mathcal{W}_e$ with respect to a given tweet e is calculated by $\text{score}(w, e) = \text{val}_e$. If $w \notin \mathcal{W}_e$, we assume $\text{score}(w, e) = 0$. Formally, the function f is defined as follows:

$$f(S) \triangleq g(S) - \ell(S) = \sum_{w \in \mathcal{W}} \sqrt{\sum_{e \in S} \text{score}(w, e)} - \sum_{e \in S} \ell_e \ , \tag{9}$$

where for the linear function ℓ two options are considered: (i) $\ell_e = |01/01/2019 - \text{T}(e)|$ is the absolute difference (in number months) between time of tweet e and the first of January 2019, (ii) $\ell_e = |\mathcal{W}_e|$ is the length of each tweet, which

enables us to provide shorter summaries. Note that the monotone and submodular function g is designed to cover the important events of the day without redundancy (by encouraging diversity in a selected set of tweets) (Kazemi et al., 2019).

The main observation from Fig. 8 is that DISTORTED-STREAMING clearly outperforms the sieve-streaming algorithm and the Greedy Dynamic Program algorithm in terms of objective value, where the gap between their performances grows for larger values of k. The utility of other offline algorithms is slightly better than that of our proposed streaming algorithm. We also see that while distorted greedy is by far the fastest offline algorithm, the computational complexities of both streaming algorithms are negligible with respect to the other offline algorithms.



(a) $\ell_e = |01/01/2019 - \mathrm{T}(e)|$ (b) $\ell_e = |\mathcal{W}_e|$ (c) $\ell_e = |01/01/2019 - \mathrm{T}(e)|$ (d) $\ell_e = |\mathcal{W}_e|$

Figure 8. Twitter text summarization: We compare algorithms for varying values of the cardinality constraint k. In figures (a) and (c) the linear cost is the difference between the time of the tweet and the first of January 2019. In figures (b) and (d) the linear cost is the number of keywords in each tweet.